CROSSTALK

May / June 2011 The Journal of Defense Software Engineering Vol. 24 No. 3

PEOPLE SOLUTIONS TO SOFTWARE PROBLEMS

Report Documentation Page

Form Approved
OMB No. 0704-0188

1. REPORT DATE	2. REPORT TYPE	3. DATES COVERED
MAY 2011		**00-00-2011 to 00-00-2011**

4. TITLE AND SUBTITLE	5a. CONTRACT NUMBER
CrossTalk. The Journal of Defense Software Engineering. Volume 24, Number 3. May/June 2011	5b. GRANT NUMBER
	5c. PROGRAM ELEMENT NUMBER
6. AUTHOR(S)	5d. PROJECT NUMBER
	5e. TASK NUMBER
	5f. WORK UNIT NUMBER
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) **517 SMXS MXDEA,6022 Fir Ave,Hill AFB,UT,84056-5820**	8. PERFORMING ORGANIZATION REPORT NUMBER
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)	10. SPONSOR/MONITOR'S ACRONYM(S)
	11. SPONSOR/MONITOR'S REPORT NUMBER(S)

12. DISTRIBUTION/AVAILABILITY STATEMENT
Approved for public release; distribution unlimited

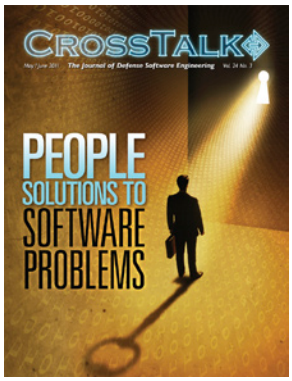
13. SUPPLEMENTARY NOTES

14. ABSTRACT

15. SUBJECT TERMS

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
a. REPORT	b. ABSTRACT	c. THIS PAGE			
unclassified	**unclassified**	**unclassified**	**Same as Report (SAR)**	**36**	

Departments



Cover Design by Kent Bingham

People Solutions to Software Problems

CrossTalk

OUSD(AT&L) Stephen P. Welby
NAVAIR Jeff Schwalb
DHS Joe Jarzombek
309 SMXG Karl Rogers

Publisher Justin T. Hill
Advisor Kasey Thompson
Article Coordinator Lynne Wade
Managing Director Brent Baxter
Managing Editor Brandon Ellis
Associate Editor Colin Kelly
Art Director Kevin Kiernan

Phone 801-775-5555
E-mail stsc.customerservice@hill.af.mil
CrossTalk Online www.crosstalkonline.org

CrossTalk, The Journal of Defense Software Engineering is co-sponsored by the Under Secretary of Defense for Acquisition, Technology and Logistics (OUSD(AT&L)); U.S. Navy (USN); U.S. Air Force (USAF); and the U.S. Department of Homeland Defense (DHS). USD(AT&L) co-sponsor: Deputy Assistant Secretary of Defense for Systems Engineering. USN co-sponsor: Naval Air Systems Command. USAF co-sponsor: Ogden-ALC 309 SMXG. DHS co-sponsor: National Cyber Security Division in the National Protection and Program Directorate.

The USAF Software Technology Support Center (STSC) is the publisher of **CrossTalk** providing both editorial oversight and technical review of the journal. **CrossTalk's** mission is to encourage the engineering development of software to improve the reliability, sustainability, and responsiveness of our warfighting capability.

Subscriptions: Visit <www.crosstalkonline.org/subscribe> to receive an e-mail notification when each new issue is published online or to subscribe to an RSS notification feed.

Article Submissions: We welcome articles of interest to the defense software community. Articles must be approved by the **CrossTalk** editorial board prior to publication. Please follow the Author Guidelines, available at <www.crosstalkonline.org/submission-guidelines>. **CrossTalk** does not pay for submissions. Published articles remain the property of the authors and may be submitted to other publications. Security agency releases, clearances, and public affairs office approvals are the sole responsibility of the authors and their organizations.

Reprints: Permission to reprint or post articles must be requested from the author or the copyright holder and coordinated with **CrossTalk.**

Trademarks and Endorsements: **CrossTalk** is an authorized publication for members of the DoD. Contents of **CrossTalk** are not necessarily the official views of, or endorsed by, the U.S. government, the DoD, the co-sponsors, or the STSC. All product names referenced in this issue are trademarks of their companies.

CrossTalk Online Services:
For questions or concerns about crosstalkonline.org web content or functionality contact the **CrossTalk** webmaster at 801-417-3000 or webmaster@luminpublishing.com.

Back Issues Available: Please phone or e-mail us to see if back issues are available free of charge.

CrossTalk is published six times a year by the U.S. Air Force STSC in concert with Lumin Publishing <luminpublishing.com>.
ISSN 2160-1577 (print); ISSN 2160-1593 (online)

CrossTalk would like to thank the OUSD(AT&L) for sponsoring this issue.

PEOPLE
Our Most Valuable Asset

One of our greatest challenges is how we approach building great teams of people and improving how we recruit, grow, and mature the systems and software engineering professionals who contribute to the nation's critical defense systems. As we continue to address the Recruit–Train–Retain objectives laid out in the 2008 National Defense Authorization Act, we must identify workforce competencies crucial for executing systems and software engineering functions within acquisition programs. In addition, we must enable realistic workforce development efforts by ensuring that these education, training, and experience requirements are balanced with job demands.

As illustrated in the featured articles, all components of the department must work together to enhance the capability and capacity of the systems and software engineering workforce through training and educational initiatives. For example, the Navy is expanding its training to meet new and evolving needs throughout its Systems Engineering Educational Continuum for Science, Technology, Engineering and Mathematics (STEM). The Navy's Systems Engineering Stakeholders Group recently conducted a Naval Systems Engineering "lessons learned" conference to develop education materials to be used in the Naval Postgraduate School and U.S. Naval Academy engineering curricula. The Naval Air Systems Command graduated its first two cohorts of the Master of Science and Systems Engineering in partnership with the Naval Postgraduate School. In addition, it established advanced degree and certificate programs in physics, mathematics, and other technical disciplines, including the Joint Executive Systems Engineering Management degree program.

The Air Force established a STEM governance structure at the three-star level and a STEM Advisory Council to address workforce requirements. The Air Force also developed a STEM strategic plan called Bright Horizons. The Air Force's Scientist and Engineer Advisory Council is evaluating the need for an initial skills training course for new Systems Engineering hires. It is investigating various strategic initiatives addressing an Air Force-wide solution for present and future science and engineering workforce capability requirements and the mechanisms for fulfilling them.

The Air Force Institute of Technology developed and implemented the Software Professional Development Program, a series of continuing education courses for the software workforce to improve software management and engineering skills. The Army employs comprehensive Individual Development Plans for all individuals in the Army Acquisition Corps and uses numerous training and educational opportunities for their current and new employees, including developmental assignments.

We cannot overemphasize that our people are our greatest asset. The department with support from Congress has made workforce development, especially in the areas of systems and software engineering and STEM, a top priority. We as individuals working in these areas must do our part to take advantage of these opportunities and encourage our colleagues to do the same. Together we can make a difference and truly improve the outcomes of our crucial acquisition programs.

Stephen P. Welby
Deputy Assistant Secretary of Defense for Systems Engineering, Office of the Under Secretary of Defense for Acquisition, Technology and Logistics

Integrating Humans With and Within Complex Systems

Challenges and Opportunities

Azad M. Madni, University of Southern California

Abstract. The integration of humans with and within today's ever more complex and increasingly more adaptive software and systems poses an ever-growing challenge. This paper discusses this integration challenge from the perspective of capitalizing on the strengths of humans, software, and systems while circumventing their respective limitations. Specific findings and examples of integration challenges that go beyond the usual human factors perspective are presented. The paper concludes with a research agenda for advancing the state-of-the-art in integrating humans with adaptive software and systems.

Introduction

The potential for "disconnect" between people and technology is well-documented in the literature for both consumer products and large scale defense, energy, and transportation systems [1, 2, 3]. The Patriot missiles deployed in the 2003 Iraq war offer an excellent illustration of this disconnect. Operators of this missile were trained to trust the system's software because the Patriot missile is a highly automated system. Such trust is essential especially when operating in a heavy missile attack environment [4]. This was not the case in the Iraqi battlespace in which the missile batteries were operating in an environment sparsely populated with missiles but with several friendly aircraft. The inadequately trained missile operators were unaware that the Patriot radar system was susceptible to recording spurious hits and occasionally issuing false alarms (i.e., mistaking friendly aircraft for enemy missiles) without displaying the uncertainty in target identification. Not surprisingly, these operators tended to trust the system's assessments and missile launch decisions against potentially hostile targets. These factors were in play in the unfortunate shoot down of a British Tornado and a U.S. Navy F/A-18. A Defense Sciences Board study concluded that, "more operator involvement and control in the function of a Patriot battery" was necessary to overcome the system's limitations [4]. Despite this recognition, system operators continue to be unfairly blamed for systemic failures. This fact did not go unnoticed by Chiles [2] who cautioned, "Too often operators and crews take the blame after a major failure, when in fact the most serious errors took place long before and were the fault of designers or managers whose system would need superhuman performance from mere mortals when things went wrong."

The primary design flaws that Chiles refers to were largely failures in proper coordination of interactions between people and technology during system development and operation [5]. In recent years, the need for systems to become increasingly more adaptive to cope with changes in the operational environment has made the integration of humans with software and systems even more challenging. In response to these challenges, the DoD made a concerted push to incorporate human considerations into the systems engineering lifecycle [6]. This emphasis led to the creation of the new multidisciplinary field of Human Systems Integration (HSI) [7, 8]. HSI is the study of interactions between humans and systems to produce human-system designs that are compatible, safe, consistent, and efficient. These interactions continue to become increasingly more complicated as human roles continue to evolve from that of an operator outside the system to that of an agent within the system. Compounding the problem is the fact that misconceptions about what it takes to integrate humans with software and systems continue to linger in the software and systems engineering communities [9]. Perhaps the single biggest misconception is that humans are "suboptimal job performers." This mindset leads to software and systems that are specifically designed to shore up or compensate for human shortcomings. With this mindset, it is hardly surprising that humans are forced to operate or work within systems that are inherently incompatible with their conceptualization of work. This paper reviews what we know about humans, discusses the consequences of unwarranted assumptions in design, and presents a HSI research agenda to advance the state-of-the-art in developing adaptive human-machine systems.

What We Know About Humans

Humans have specific strengths and limitations that need to be well-understood before determining how best to integrate them with software and systems [10, 11, 12, 13]. The key findings from the literature that bear on human-system integration are:

- **Human Performance:** [14, 15, 16, 17]
  -Varies nonlinearly with several factors
  -Follows an inverted U-curve relative to stress
  -Excessive cognitive complexity can lead to task shedding and poor performance [14]
- **Human Error:** [14, 16]
  -Lack of inspectability into system operation can induce human error
  -Incompatibility between human processes and machine algorithms can lead to human error
  -Sustained cognitive overload can lead to fatigue and human error
- **Human Adaptivity:** [18, 19]
  -Adaptivity is a unique human capability that is neither absolute or perfect
  -Humans do adapt under certain conditions but usually not quickly
  -Human adaptation rate sets an upper bound on how fast systems can adapt
  -Tradeoff between human adaptation rate and error likelihood
  -Need to define what is acceptable error rate (context-dependent)

- **Multitasking:** [18, 19]
  -Humans do not multitask well
  -Stanford University's research findings show that so-called high multitaskers have difficulty filtering out irrelevant information, can't compartmentalize to improve recall, and can't separate contexts
- **Decision Making Under Stress:** [18, 19]
  -Under stress humans tend to simplify environment by disregarding/underweighting complicating factors
  -Reduced ability to process multiple cues or perform tradeoffs
- **User Acceptance:** [14, 18, 19]
  -Overly complex system design can lead to rejection of the system
  -Humans do not have to really understand software/system operation to develop confidence and trust in system
- **Risk Perception and Behavior:** [20, 21, 22, 23, 24]
  -Humans accept greater risks when in teams
  -Humans have a built in target level of acceptable risk
- **Human-System Integration:** [9, 25, 26]
  -Humans are creative but rarely exactly right; however, human errors usually tend to be relatively minor
  -Software/system solutions tend to be precisely right, but when wrong they can be way off

The literature on human-machine systems offers ample evidence that poorly designed automation can produce performance degradation of the overall human-machine system. An important aspect of such performance degradation is the lack of "fit" between the mental models of humans, cognitive demand of the work environment, and automation design.

Poor Automation Design Can Degrade Human Performance

- **Cognitive Load in Supervising Automation:** [27, 28]
  -The cognitive load when monitoring automated task performance can outweigh potential automation benefits
- **Automation-induced Complacency:** [29]
  -Over-reliance on automation can increase errors as humans begin to rely on automated cues rather their own vigilant information seeking and cognitive processing [30]
- **Partially Automated System with Incomplete Knowledge:** [31]
  -The system, operating outside its competence regime, stays in the loop and continues to critique operator performance based on erroneous assessment of work constraint violations
- **Mistrust of Automation:** [1]
  -Can lead to disuse, neglect, underutilization
  -Typically arises from poor design (e.g., high rate of false alarms in an alerting system)
- **Erosion of Operator's Expertise and Engagement:** [32]
  -Inappropriate automation can lead to skill decay or dysfunctional skills
  -Operator can no longer intervene effectively when automation malfunctions

Unwarranted Assumptions in Design Can Produce Unintended Consequences

System designs are often based on unstated and occasionally unwarranted assumptions about human behavior. These assumptions can often lead to unintended consequences and give rise to systemic failures. The following paragraphs offer examples of unexpected outcomes and unintended consequences that can be traced to unwarranted assumptions about human behavior:

Risk Homeostasis: Wilde specifically hypothesized that humans have a target level of acceptable risk (that typically varies among humans but is fixed for each individual). He called this risk homeostasis. He argued that safety features and campaigns tend to shift rather than reduce risk. While initially subject to criticism, this hypothesis was confirmed through studies in Munich, Germany, and in British Columbia, Canada. In the Munich study, half a fleet of taxicabs was equipped with anti-lock brakes (ABS), while the other half was provided conventional brake systems. Pursuant to testing, it was discovered that the crash rate was about the same for both types. Wilde concluded that this result was due to the fact that drivers of ABS-equipped cabs took more risks because they assumed that the ABS would provide the requisite protection in hazardous driving conditions. By the same token, the non-ABS drivers drove more carefully because they recognized that they were driving without an ABS system and had to be more careful in hazardous driving conditions.

Design-induced Human Error: In 2008, a Metrolink commuter train crashed headlong into a Union Pacific freight locomotive after going through four warning lights. The engineer (i.e., the driver) failed to hit the brakes before the train crashed. A teenage train enthusiast later claimed to have received a cell phone text message from the driver a minute before the collision [3].

So, was the Metrolink train accident a human error, a systemic problem that manifested itself as a human error, or both? The answer is BOTH. Since the driver was doing a split-shift, he was clearly tired. He was also multitasking. Humans don't multitask well and are error-prone in such circumstances. However, the system was also not designed for integration with the human in that the system design assumed an optimal human i.e., one who could multitask, one who would not fatigue, and one who was goal-driven and a utility maximizer. Humans are not any of these! This was an accident waiting to happen [9].

Human Role-Architecture Mismatch: The human role in relation to the system or within the system plays a significant role in both system architecture design and algorithm selection. For example, if the human is expected to be replaced by automation in the future, then the system architecture would emphasize a different set of quality attributes than if the human role was integral to the system (i.e., permanent). The same is true of algorithm selection. Consider the selection of a route planning algorithm for an autonomous ground vehicle. Invariably, a constrained optimization algorithm would be used to solve the route planning problem. Now consider route planning for a human-supervised ground vehicle in which the human needs to specify waypoints along the way. In this case, the algorithm needs to be interactive, inspectable, and understandable so that the human can intervene to specify waypoints. As such, a heuristic algorithm becomes preferable to the optimization algorithm because the heuristic algorithm

allows the human to understand system reasoning and intervene effectively [33]. In this example, algorithm inspectability is more important than algorithm optimality.

Indiscriminate Automation: Roughly a decade ago, a blind side indicator was developed for automobiles to show an object in the driver's blind side. This device was never approved, because behavioral research showed that drivers were going to overuse the indicator, and no longer bother to look back over their shoulder when changing lanes. This would have been clearly an undesirable change in driver behavior. The lesson clearly is that indiscriminate use of technology without understanding its impact on human behavior patterns can potentially change human behavior, and not necessarily for the better. This kind of analysis is key to avoiding unintended consequences [33, 34].

The foregoing examples provide several key insights. First, in a tightly coupled system, any change to the machine will cause humans to change as well. Such a change could be undesirable in the sense that it could lead to unintended consequences. Second, unwarranted assumptions about the human can lead to tragic accidents [33]. For example, assuming that humans are optimal information processors can lead to dire results because humans do fatigue and don't multitask well. Third, the role of the human in the overall system is key to architectural paradigm and algorithm selection. Specifically, it is important to determine whether the human is central to system operation, or merely an adjunct or enabler to be replaced by automation in the future. Fourth, system architects need to focus on combined human-system performance, not the performance of each in isolation. This also means that the focus should be on combined metrics, not individual metrics. And, finally, a change in the operational environment can potentially change how people perceive and compensate for risks [9].

HSI Research Agenda

Figure 1 presents a HSI research framework for investigating high payoff research opportunities. As shown in this framework, HSI research needs to address human capabilities and limitations, evolving human roles, system adaptation contexts, and the systems engineering lifecycle.

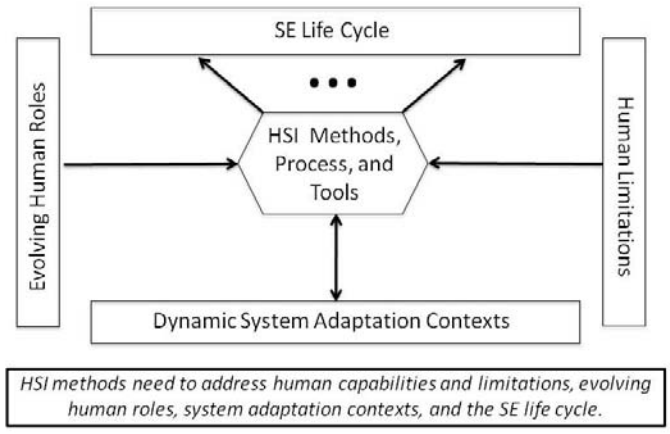


Figure 1: High Payoff HSI Research

This framework, in part, is inspired by the recently completed DoD-sponsored National Research Council (NRC) study [8], which recommended: (a) the development of shared representations to enable meaningful communications among hardware, software and HSI designers as well as within the human-system design group, and within the stakeholder community; (b) the extension and expansion of existing HSI methods and tools including modeling and simulation methods, risk analysis, and usability evaluation tools; and (c) the full integration of humans with engineered systems. In light of these recommendations, several research thrusts need to be pursued before developing HSI methods, processes, and tools for infusing HSI considerations into the software and systems engineering lifecycle. These research thrusts are discussed next:

Methodology for HSI Problem Identification: The underlying HSI problems could be one or more of the following: system is too difficult to operate, human error rates are unacceptably high, system is not being used or is not being used as intended, system is too hard to maintain, system is too expensive, and system does not scale. To this end, research is needed in advancing the state-of-the-art in concept engineering, virtual prototyping, interactive human-system simulations, human behavior and performance modeling, behavioral informatics, and synthetic environments that capture the geospatial and socio-cultural characteristics of the operational environment.

Development of a Shared Representation: In keeping with the NRC's recommendation, the development of a shared representation is key to enabling meaningful communication and collaboration among hardware engineers, software engineers, HSI personnel, and the larger stakeholder community. To this end, the development of a common ontology and a lexical data base that eliminates the polysemy and synonymy problems among the different disciplines can serve as a sound starting point.

Expansion of Existing Methods and Tools: Existing modeling and simulation tools as well as risk analysis and usability evaluation methods have focused on front-end analysis with a narrow view of human-system integration [8]. Research is needed to extend the methods, processes, and tools to span the full software and system lifecycles while also expanding the scope of the modeling, simulation and analysis tools to address human integration with adaptive systems.

Human Performance Modeling: Human performance varies nonlinearly with a variety of factors such as stress, anxiety, workload, fatigue, and motivation levels. For example, the Yerkes-Dodson law shows that as stress increases, so does performance, up to a point beyond which it rounds out and starts decreasing (the well-known inverted U-Curve). Cognitive workload becomes a key concern in several mentally taxing functions/jobs [10, 11, 35] such as anesthesiology, air traffic control, military command and control, and nuclear power plant operation. The key characteristics of high cognitive load tasks are that they are stimulus-driven (i.e., not self-paced), they produce large fluctuations in demand, they involve multi-tasking, they generate high stress and, they tend to be highly consequential [9]. Research is needed in developing adaptive human performance models and simulations that are sensitive to the various factors

that affect performance. Such models can then be used to "test drive" and evaluate candidate designs from an HSI perspective.

Architecture Design: The architectural design of adaptive human-machine systems is highly dependent on the roles that humans play and the transition between roles in the overall adaptive system. In particular, human roles have a significant impact on the architecture depending on whether the human is central to the system, a monitor of the system with override privileges, or merely an enabling agent [9]. Research is needed in adaptive architecture design with various levels of human involvement in system operation. In particular, a human performance testbed needs to be developed that can support architecture sensitivity analysis to changes in critical human and environmental parameters and architecture adaptation in response to changes in these parameters.

Consolidating Human Performance Body of Knowledge: At the present time, the body of knowledge in human performance is highly fragmented. Exemplar categories include: human adaptivity contexts and rates; workload (cognitive and psychomotor); decision making (under time-stress, uncertainty, and risk); risk perception and risk homeostasis; socio-cultural factors in decision making, negotiations, and consensus building; vigilance and arousal; and physiological/mental stress, and fatigue. Research is needed to determine where and how these various considerations interact and then to consolidate the body of knowledge with use cases that reflect the needs of systems engineers, software engineers, and HSI practitioners.

Integrated Aiding-Training Continuum: Recent research has shown that aiding and training lie along a human performance enhancement continuum [12]. Research is needed in defining adaptive architectures for integrated aiding and training, capable of dynamically repurposing content (e.g., Shareable Content Objects) for aiding, training, and performance support based on user needs and the operational context [35].

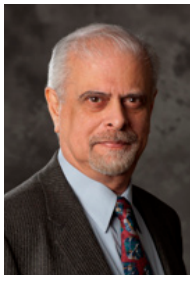
HSI Patterns: Humans interact with systems differently based on their role (i.e., supervisor, monitor, enabler) relative to the system. Human-system interaction for each role and transition between roles tends to be different and potentially amenable to characterization through patterns. For example, the transition of human role from a supervisor to an enabler based on changes in context can be characterized by a pattern. Research is needed in defining the adaptation requirements of various types of architectures based on role transitions and capturing these findings in the form of HSI architectural patterns.

Conclusions

The systems acquisition and engineering communities have recently began to focus on addressing human capabilities and limitations and their implications on the design, operation and maintenance of complex systems. The discipline of HSI is intended to remedy this problem. However, for HSI to make inroads into the systems acquisition and engineering communities, several advances need to occur. First, the fragmented body of knowledge in human performance needs to be consolidated, expanded, and transformed into a form that lends itself to being incorporated into software and systems engineering practices. Second, the HSI community needs to make the business case to communicate the value proposition of HSI in lifecycle cost reduction to the system development community. Third, systems acquisition and systems engineering policies need to be appropriately revised to reflect the inclusion of HSI principles and guidelines.

The specific approaches by which these recommendations can be implemented are as follows. Initially, use case scenarios need to be defined to frame the relevant contexts for the system acquisition and engineering communities. Next, a flexible, open, process-oriented, systems engineering tool (preferably in use within the DoD) with library facilities needs to be selected. This tool can serve as a convenient starting point for consolidating human considerations and incorporating HSI processes into the software and systems engineering lifecycles. The tool should support multiple lifecycle models (e.g., incremental prototyping, evolutionary development, etc.). The tool needs to incorporate a library of principles from the behavioral and social sciences, as well as from human factors engineering. In this regard, the key issues identified in this paper need to be addressed along with their impact on performance, cost, and schedule. Finally, an end user oriented front-end should be provided to the tool to avoid the need for an intermediary. With such a methodology and toolset, it will eventually become possible to convey the value proposition of addressing HSI considerations early and throughout the software/system lifecycle to the system acquisition and engineering communities, while also assuring end user acceptance of the tool.

ABOUT THE AUTHOR



Dr. Azad Madni is a Professor and Director of the Systems Architecting and Engineering Program in the Viterbi School of Engineering at the University of Southern California. He is also the founder and Chief Scientist of Intelligent Systems Technology, Inc. He received his B.S., M.S., and Ph.D. degrees in engineering from UCLA. His research has been sponsored by DoD, NIST, DHS S&T, DTRA, NIST, DoE, and NASA. He is an elected Fellow of IEEE, INCOSE, SDPS, IETE, and an Associate Fellow of AIAA. He is listed in the major Marquis' Who's Who, including Who's Who in America.

Phone: (213) 740-3442
E-mail: azad.madni@usc.edu

REFERENCES

1. R. Parasuraman and V. Riley, Humans and automation: Use, misuse, disuse, abuse, Human Factors 39 (1997), 230-253.
2. J.R. Chiles, Inviting Disaster: Lessons from the Edge of Technology, Collins, New York, 2001.
3. S. Hymon, Metrolink Report Urges More Oversight, Safety Equipment, Los Angeles Times, January 8, 2009.
4. Defense Science Board, Defense Science Board Task Force on Patriot System Performance, Report Summary DTIC No. ADA435837, January 2005.
5. D.D. Woods, and N.B. Sarter, "Learning from automation surprises and going sour accidents," in N. Sarter and R. Amalberti (Editors), Cognitive Engineering in the Aviation Domain, Erlbaum, Hillsdale, NJ, 2000, pp. 327-353.
6. Department of Defense Handbook, Human Engineering Program Process and Procedures, MIL-HDBK-46855, January 31, 1996.
7. H.R. Booher, R. Beaton, and F. Greene, "Human Systems Integration," in A. Sage and W.B. Rouse (Editors), Handbook of Systems Engineering and Modeling, John Wiley and Sons, Hoboken, NJ, 2009, pp. 1319-1356.
8. National Research Council, "Human-system integration in the system development process: A new look," Committee on Human-System Design Support for Changing Technology, in R.W. Pew and A.S. Mavor (Editors), Committee on Human Factors, Division of Behavioral and Social Sciences and Education, National Academies Press, Washington, D.C., 2007.
9. A.M. Madni, "Integrating Humans with Software and Systems: Technical Challenges and a Research Agenda," INCOSE Journal of Systems Engineering, Vol. 13, No. 3, 2010.
10. A.M. Madni, HUMANE: A designer's assistant for modeling and evaluating function allocation options, Proceedings of Ergonomics of Advanced Manufacturing and Automated Systems Conference, Louisville, KY, August 16-18, 1988b, pp. 291-302.
11. A.M. Madni, HUMANE: A knowledge-based simulation environment for human-machine function allocation, Proceedings of IEEE National Aerospace & Electronics Conference, Dayton, Ohio, May, 1988c.
12. A.M. Madni, The role of human factors in expert systems design and acceptance, Human Factors Journal 30 (1988a), 395-414.
13. D. Meister, Conceptual aspects of human factors, The Johns Hopkins University Press, Baltimore, MD, 1989.
14. Madni, A.M., and Moses, F. "An Intelligent Soldier-Vehicle Interface for Future Close Combat Vehicles," Proceedings of 1985 IEEE International Conference on Systems, Man, and Cybernetics, Tucson, Arizona, November, 1985, pp. 754-756.
15. T.B. Sheridan, Man-machine systems, MIT Press, Cambridge, MA, 1974.
16. C.D. Wickens and J.G. Hollands, Engineering psychology and human performance, Pearson, Canada, 1999.
17. R.M. Yerkes and J.D. Dodson, The relation of strength of stimulus to rapidity of habit formation, Journal of Comparative Neurology and Psychology 18 (1908), 459-482.
18. Madni, A.M. "Integrating Humans with Software and Systems: Technical Challenges and a Research Agenda," INCOSE 2010 LA Mini-Conference, Loyola Marymount University, October 16, 2010.
19. Madni, A.M. "Integrating Humans with Software and Systems: Technical Challenges and a Research Agenda," 22nd Annual Systems and Software Technology Conference, Salt Lake City, Utah, April 27, 2010.
20. P. Slovic and A. Tversky, Who accepts savage's axiom? Behavioral Science 19 (1974), 368-373.
21. M.A. Wallach, N. Kogan, and D.G. Bern, Group influence on individual risk taking, Journal of Abnormal and Social Psychology 65 (1962), 75-86.
22. M.A. Wallach, N. Kogan, and D.G. Bern, Diffusion of responsibility and level of risk-taking in groups, Journal of Abnormal and Social Psychology 68 (1964), 263-274.
23. J.A.F. Stoner, Risky and cautious shifts in group decisions: The influence of widely held values, Journal of Experimental Social Psychology 4 (1968), 442-459.
24. G.J.S. Wilde, Target risk 2: A new psychology of safety and health: What works? What doesn't? And why? PDE Publications, Toronto, 2001.
25. K.R. Hammond, R.M. Hamm, J. Grassia, and T. Pearson, Direct comparison of the efficacy of intuitive and analytic cognition in expert judgment, IEEE Transactions on Systems, Man and Cybernetics 17 (1987), 753-770.
26. K.R. Hammond, Human judgment and social policy: Irreducible uncertainty, inevitable error, unavoidable justice, Oxford University Press, New York, 1996.
27. A. Kirlik, Modeling strategic behavior in human-automation interaction: Why an "aid" can (and should) go unused, Human Factors 35 (1993), 221-242.
28. T.B. Sheridan, Telerobotics, automation, and human supervisory control, MIT Press, Cambridge, MA, 1992.
29. R. Parasuraman, R. Molly, and I.L. Singh, Performance consequences of automation-induced complacency, The International Journal of Aviation Psychology 3 (1993), 1-23.
30. K.L. Mosier and L.J. Skitka, "Human decision makers and automated aids: Made for each other?" in R. Parasuraman and M. Mouloua (Editors), Automation and human performance: Theory and applications, Lawrence Erlbaum Associates, Mahwah, New Jersey, 1996, pp. 201-220.
31. S.A.E. Guerlain, Factors influencing the cooperative problem-solving of people and computers, Proceedings of the Human Factors and Ergonomics Society 37th Annual Meeting, Santa Monica, CA: Human Factors and Ergonomics Society, 1993, pp. 387-391.
32. G. Klein, Implications of the Naturalistic Decision Making Framework for Information Dominance, Report No. AL/CF-TR-1997-0155, Wright-Patterson AFB, OH, Armstrong Laboratory, Human Engineering Division, 1997.
33. A.M. Madni, A. Sage, and C.C. Madni, Infusion of cognitive engineering into systems engineering processes and practices, Proceedings of the 2005 IEEE International Conference on Systems, Man, and Cybernetics, October 10-12, 2005, Hawaii.
34. A.M. Madni, and S. Jackson, Towards a conceptual framework for resilience engineering, IEEE Systems Journal 3 (2009).
35. A.M. Madni, "Towards a Generalizable Aiding-Training Continuum for Human Performance Enhancement," INCOSE Journal of Systems Engineering, Volume 14, Number 1, 2011.

Improving Software Engineering Through Holistic Project Coaching

Dr. Randall Jensen
Software Organizational Development Office
Fred Smullin
Woodbury Technologies, Inc.
Joyce Peters
OO-ALC/DPD Force Development Division
Kasey Thompson
Software Organizational Development Office
Dr. Doretta E. Gordon
Southwest Research Institute

Abstract. The process to successfully engineer quality software suffers from challenges identified over 40 years ago. Upon deeper review, a majority of the factors related to software development failure are human factors. Including a Project Coach (PC) who is focused on humanistic issues in the software engineering process will have a positive impact in addressing software engineering challenges. A PC focuses on Knowledge Management (KM), cyclical assessment, informal learning, and dynamics coaching to ensure team harmony and growth, sound project management practices, and most importantly–quality, on time software.

The 1968 NATO International Software Engineering Conference [1] in Munich, Germany, raised a series of complaints about computer software including its unreliable nature, late delivery, cost-prohibitive nature of modification, challenges in maintenance, inadequate performance, and budget cost excesses. This conference resulted in the coining of the phrase "software engineering." Over 40 years later, the software engineering field has failed to significantly diminish or eliminate many of these serious complaints.

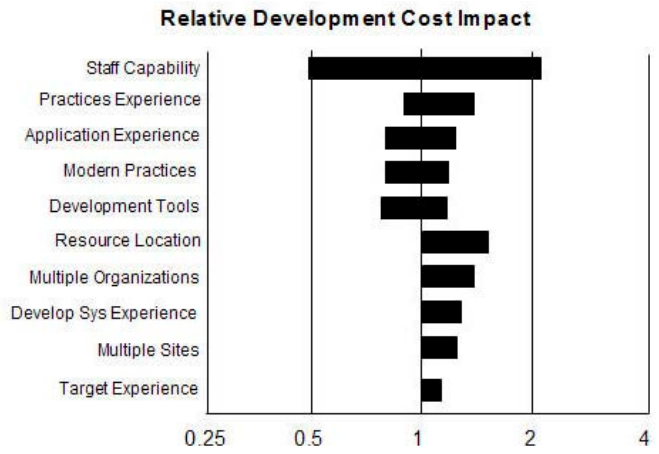
An IEEE Spectrum article entitled "Why Software Fails" [2] cites 12 common failure factors:

1. Unrealistic or unarticulated project goals
2. Inaccurate estimates of needed resources
3. Badly defined system requirements
4. Poor reporting of the project's status
5. Unmanaged risks
6. Poor communication among customers, developers, and users
7. Use of immature technology
8. Inability to handle the project's complexity
9. Sloppy development practices
10. Poor project management
11. Stakeholder politics
12. Commercial pressures

While the article categorized deficiencies into technical, project management, and business decision deficiencies, it can also be noted that nine of the 12 factors are human factors. A recent focus has been placed on management and people issues in the software development process.

This focus on the importance of people in the software engineering process has appeared in early works ranging from McGregor's [3] Theory X - Theory Y to Deming's [4] Total Quality Management approach. To further quantify the impact of people in the software development process, both the Constructive Cost Model [5] and Software Evaluation and Estimation of Resources (SEER) [6] software estimation models forecast the relative impact of the development environment parameters. The most important parameter group (Staff Capability) shows a relative cost impact of 0.5 on the positive side and greater than 2.2 on the negative side (see Figure 1).

Figure 1: Relative cost impact of the Constructive Cost Model and SEER environment parameters



A second aspect relative to the importance of people in the product–process–people triad is the generational diversity that is now representative of a majority of the workforce population. Today's workforce is divided among Baby Boomers; generally described as those born between 1944 and 1966, Generation X; born between 1967 and 1979, through Generation Y; born between 1980 through 1995. Each workforce generation is shaped by key events in their development and results in differing needs, desires and expectations as it relates to work environments. For example, Generation Y has never known a time in which there was no Internet. They have grown up in a "connected" world. This has helped to shape their expectations in terms of teaming, communication, learning and information sharing. There are growing cases of Generation Y Project Managers (PMs) who are now managing Baby Boomers [7]. These dynamics presents new challenges to projects such as software development and, at a minimum, require an awareness of differences in expectations and outlooks.

Unless people are considered as an equally important leg supporting the product–process–people triad of software engineering, the results will remain inconsistent and unstable at best.

1. Holistic Project Coaching

Given this background, one approach which has proven to have a positive impact on the software development process is Holistic Project Coaching (HPC). HPC is an experiential, performance-oriented development process that builds a project team's capability to achieve short- and long-term project success. It is conducted via individual and team-based interactions, incorporates multiple perspectives, and focuses on building positive actions based on mutual trust and respect.

HPC utilizes an existing member of the project team to act as the PC. The PC is not an additional resource; rather they are an internal team member who works in conjunction with the PM to ensure project success by bridging technical and non-technical issues as was the case in our case studies. The PM and PC are co-supportive of each other. The PC supports the PM with Project Management Body of Knowledge (PMBOK) practices and techniques, but primarily focuses on such activities as conflict management, purposeful on-the-job education, cyclical assessment and Human Performance Technology (HPT); all of which are outside the PMBOK scope. This allows the PM to continue his or her focus on development, schedule, and quality.

HPC is applied to both individuals and the team as a whole. The HPC process has four primary underpinnings that serve as the foundation for HPC activities. These include KM, cyclical assessment, informal learning, and dynamics coaching. The PC is responsible for the facilitation and application of the following four prongs of HPC:

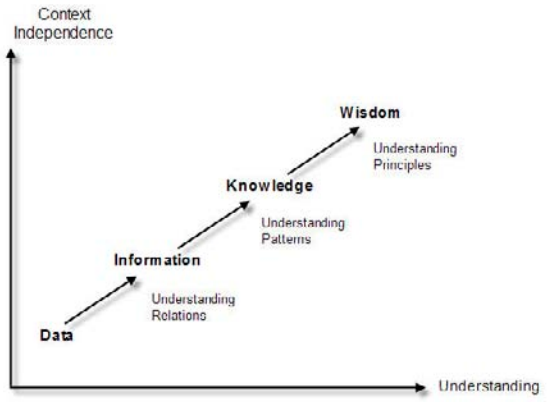
1.1 HPC and KM

The objective of KM is to improve the quality of decision making by ensuring that the right information is available to the right person, at the right time, to enable an individual or team to make an informed decision. The quality of services delivered is directly impacted by how the team responds to circumstances. Their response will be governed by what they perceive their options to be as well as the consequences and benefits of those options. Their overall knowledge of the situation will ultimately influence how they execute processes and what the output quality will be.

KM can aid the team's ability to adapt and overcome challenges through collaboration and shared knowledge. An effective KM strategy will positively influence project costs by increasing staff capabilities through knowledge discovery, sharing, and collaboration thus contributing to workforce development.

As shown in Figure 2, KM is often expressed as a knowledge hierarchy [8]. Data is the foundation from which we obtain information, build knowledge, and apply wisdom. Data itself is just a point in space and time without reference to either space or time. It has no context and thus has little or no meaning. Information (the what, who, when, where) is built from the understanding of the relations between the data. Information generally does not provide a foundation for why the data exists, what it is, nor how the data is likely to change over time. Thus information is simply the relationship between data and has great dependence on context for its meaning with little implication for the future.

Figure 2: Data – Information – Knowledge – Wisdom Hierarchy



Knowledge (the how) is based on patterns that exist amidst the data and information. These patterns have a tendency to create their own context rather than being context dependent like information. They also provide a high level of reliability or predictability as to how the pattern will evolve over time. They have completeness to them that information simply does not contain. Wisdom (the why) is generated when understanding of the foundational principles responsible for the patterns representing knowledge is achieved. Likewise, wisdom, even more so than knowledge, tends to create its own context.

Why is this important for the PC? The goal of KM is to try to maximize the value of knowledge holdings within an organization. This includes knowledge that is both explicit (e.g., codified) and tacit (i.e., "know how"). KM facilitates access to knowledge including pointers to tacit knowledge and thereby encourages collaboration, innovation, and promotion of existing knowledge as a foundation for new ideas.

For the PC, KM hinges on creation of a sound KM plan. A good KM program facilitates capture during the entire project lifecycle. At a minimum, a KM plan should include:

	Information	Instrumentation	Motivation
Environment Supports	**Data** Relevant and frequent feedback about the adequacy of performance. Description of what is expected of performance. Clear and relevant guides to adequate performance.	**Resources** Tools and materials of work designed scientifically to match human factors (e.g. databases, digital systems, knowledge management tools).	**Incentives** Adequate financial incentives made contingent on performance. Non-monetary incentives. Career development opportunities.
Person's Repertory of Behavior	**Knowledge** Scientifically designed training that matches the requirements of exemplary performance. Informal learning opportunities.	**Capacity** Flexible schedule of performance to match peak capacity. Selection. Supportive devices (e.g. prosthesis, text readers)	**Motives** Assessment of people's motives to work. Recruitment of people to match the realities of the situation.

- **Mission:** What are your goals? What knowledge is useful to that mission?
- **Competition:** How are you gaining and maintaining competitive advantage? How are you going to improve comprehension and knowledge building within your stakeholder community?
- **Performance:** How are you going to deliver results? How do you get the right information to the right person at the right time to improve decision making?
- **Change:** How will you cope with change? How do you make outside external knowledge available to help your organization adapt and overcome?

A critical task for the PC is facilitating knowledge transfer. Knowledge transfer is defined as the process through which one unit of an organization is affected by the experience of another. The knowledge transfer process consists of identifying the knowledge holders within the organization, motivating them to share, designing a sharing mechanism to facilitate the transfer, executing the transfer plan, measuring to ensure the transfer, and applying the knowledge transferred. The effectiveness of knowledge transfer can be measured by how the receiving organizational behavior changes; that is, how they apply it.

Some common impediments to knowledge transfer that HPC can help resolve are: areas of expertise identification, internal conflicts (territorial), generational differences, union-management relations, incentives, geography or distance, culture, knowledge visualization, faulty information, motivational issues, and lack of trust.

Tools at the PC's disposal for KM and transfer range from content and document management systems for explicit knowledge to newer web tools such as social networking sites for tacit knowledge. A robust KM strategy will most likely consist of a mix of several components that may or may not be technology based. The PC can help to determine the best mix of tools and which will best support the project team.

1.2 HPC and Cyclical Assessment

A primary role of the PC is to continuously measure the pulse of the individuals and the project team as a whole (including the PM). As previously noted, the PC is concerned with the human factor; that is, the support required to ensure success in terms of human performance. In contrast and as the name implies, a PM is primarily concerned with managing such aspects of the project as schedule, budget, and quality.

PCs and PMs are both constantly monitoring; however the PM primarily monitors the project and the PC primarily monitors the people. Additionally, the PC often includes the PM's needs in the measurement of the health of the project since the PM has a profound impact on the success of the project.

Teams are dynamic and are in the midst of a dynamic activity called a project. Because of this dynamic nature, the PC must conduct ongoing assessments of the team and individual's health and needs. But what do they measure?

The field of HPT provides several models that can help to guide the HPC in this matter. Thomas Gilbert's Behavioral Engineering Model (BEM) [9] proposes that it is possible to engineer worthy performance. To do so, the PC must provide environmental support and individual support.

Within the environment, the PC must measure what deficits exist in terms of data, resources, or incentives. Once a deficit is identified, the PC's role is to modify the environment to provide the necessary support. This can be a challenging undertaking and years of measurement using the BEM have revealed that challenges to exemplary performance in the workplace are most frequently tied to deficits in environmental support. In the words of HPT pioneers Rummler and Brache, "If you pit a good performer against a bad system, the system will win almost every time" [10].

Within the individuals, the PC must measure possible deficits in knowledge, capacity, or motivation. A deficit in knowledge is most often tied to a need for formal or informal learning. A deficit in capacity is most often tied to improving the selection process to bring on the right people for the work at hand. A deficit in motivation is probably the most challenging as this relates to intrinsic motivation within the individual. Addressing this deficit is often accomplished by merely assessing what motivates individuals and feeding that information into incentives within the environment.

A primary strength of HPC is the dynamic nature of the assessment–implementation cycle the coach uses to identify and measure the ongoing needs of the project team and to implement solutions. From each assessment, the PC can then create micro-implementation plans to meet immediate human and resource needs. Measurement tools are used from a broad array of disciplines ranging from psychology (e.g., personality tools) to organization behavior (e.g., 360-degree feedback). Once a plan is implemented, a cyclical series of assessments followed by modified implementation and reassessment continues throughout the life of the project. A key to success for the PC (and for the team) is this constant measurement and implementation cycle.

1.3 HPC and Informal Learning

One of the strengths of a project team is the ability to share knowledge and wisdom informally and to create new knowledge. This process is known as informal learning and is a primary tenet of HPC.

Of the three generations in the workforce, Generation Y has strongly embraced informal learning using techniques and tools such as blogs, Tweets, and social networks. Generation Y has never known a world without computers [11]. Similarly, a majority of Generation X has grown up in a work environment in which computer use has grown exponentially [12]. This technologically savvy workforce has different communication expectations than its Baby Boomer predecessors. Informal learning allows use of newer technology-based tools to assist in augmenting the communication and learning environment.

Traditional training tends to be monolithic in nature where pre-planned courses that consist of defined lessons and topics are prescribed for any person with a knowledge deficit in that particular area. From an organizational point of view, traditional training is efficient in terms of measurement and tracking. Learners sign up for specific courses, receive scores upon completion, and are assigned further courses based on the

outcomes. But from the individual perspective, traditional training approaches cannot easily account for individual differences, individual learning styles and preferences, and nuances in learning needs.

In contrast, informal learning is micro in nature. It is a ground-up approach. Individuals share their thoughts and experiences on specific topics. Persons interested in learning about the specific topic take in the information provided, critically assess the value and validity of the information, and assimilate the new information into their internal mental schema. This is a fundamentally different approach to learning. It is driven and directed by the learner seeking data, information, knowledge, and wisdom. Additionally, the learner must then assess the value of the new information and determine if and where this information should now exist to expand their worldview.

As such, there are several benefits to informal learning:

- **Just-in-time:** Learners seek out information from immediate colleagues, recognized experts, and easily accessible information at their point of need. It is not pre-scheduled.
- **Just What I Need:** Informal learning is efficient because learners only seek out what they need to know to meet their immediate need.
- **Gestalt:** Informal learning tends to happen in synergistic relationships and therefore often results in the creation of new knowledge. The learner's mental model now has new connections and becomes deeper and richer.
- **Critical Thinking:** The process of learning informally requires the application of critical thinking skills. Because there is no official vetting process, learners must conduct an evaluation of incoming information to determine its validity and how to assimilate it into their existing mental models.

Three primary challenges related to informal learning include:

- **Tracking and Measuring Learning:** from an organizational perspective, measurement of learning is key to determining if knowledge and skill gaps are being filled. Measuring the impact of informal learning is often more subtle; did the learner progress through the process because they found the information they needed? How long did they have to search to find the information they needed?
- **Required Communication and Collaboration Tools:** the ability to participate in informal learning is dependent upon being able to access data, information, knowledge, and wisdom. It is also closely tied to open communication channels and tools that support collaboration. Social networking and web 2.0 tools [13] are coming into accepted use at an amazing rate within organizations. These tools are ideal to both capture micro-learning content and to share it in unobtrusive manners. Common tools in this genre include blogs, wikis, tweets, crowd sourcing, and status updates.
- **Quality of Content:** A major challenge in the world of informal learning is the responsibility of the learner to make a critical determination about the quality of the content they are learning. New tools are emerging (e.g., rating systems, expert profiles systems) and new research is being conducted that will help learners make these determinations, but the responsibility still rests on the learner's ability to make a proper assessment.

As the value of informal learning becomes recognized and as new tools come to use, these challenges can be overcome.

1.4 HPC and Dynamics Coaching

As noted previously, a PC is created to focus on the humanistic needs of the team in relation to the project and provide the care and feeding needed as projects and teams progress through the project lifecycle as well as the many facets of team dynamics. For discussion purposes we make an underlying assumption that teams are composed of members selected primarily because of technical capabilities. The PC looks further to discover typically untapped, intangible resources residing within each employee. Some examples include abilities such as leadership, communication, problem solving, organization, relationship building, and consensus building to name but a few.

Many personality and strength identification tools exist to assist a PC in discovering these hidden gems. While specific tools cannot be recommended in this article due to legal restrictions, it is recommended that a combination of personality and strength indicator tools be used to provide a well-rounded data set. The personality tool reveals information about who the person is and how they prefer to work and interact. The strength tool identifies skills and talents that even the individual may not know they possess.

Once the team has identified personality and strength factors, a team-blending meeting is held where team members simply share their results and make each other aware of what each person has to offer. The PC facilitates this meeting and begins to strategize with the team concerning optimal usage of each member's strengths. More seasoned coaches can build Integration Charts where team members are combined for specific tasks based upon combined strengths or to compliment an identified limitation. An example might be to assign a team member with an innate ability to communicate with people and get them to feel comfortable in speaking and exploring issues and link them with another team member who possesses organizational and strategic thinking to meet with customer groups in building a requirements document or a risk management plan. The entire purpose of understanding the individual composition of the team is to place people in a position of strength and aptitude to lend to overall team and project success.

Redistributing tasks based on individual strengths is a Dynamics Coaching element that results in increased team productivity. This is because teams begin to grow in confidence due to the fact members are now working in areas where they are naturally talented. They also begin to blend and work more harmoniously because teams now better understand each other's strengths and recognize differences among themselves as alternate strengths rather than discordant traits which can break teams down. Individuals also show a significant increase in productivity. This may seem obvious, but Gallup polls report [14] only 32% of U.S. workers utilize their primary strengths in the work they perform daily. This statistic reveals the heightened need for an increased focus on individual and team strengths if an organization is to harness the most productivity from a team. Gallup also reports: "People who use their strengths every day are *six* times more likely to be engaged on the job and three times more likely to be happier with their lives in general. Not

only do engaged workers stay on the job longer–saving millions in training and turnover costs–but they also get more done while they are there. So when workers are able to apply their talents and strengths at work, productivity also rises" [15].

Another beneficial result of the Dynamics Coaching element is an increase in team trust. Trust is the high-octane fuel that really makes team engines roar. Steven H.R. Covey reports [16] trust increases the speed of business and reduces costs. A study published in the European Journal of Work and Organizational Psychology concluded: "Cooperative behaviors were the second strongest component of trust" [17]. Conversely, breakdowns in team civility will reduce productivity and increase costs. A 2009 national study [18] consisting of a large diverse sample of managers and employees reports that of those among coworkers who have been offended:

- **48% intentionally decreased work effort**
- **47% intentionally decrease time at work**
- **38% decreased time at work**
- **80% lost time worrying about the incident**
- **63% lost time avoiding the offender**
- **66% said their performance declined**
- **78% said their commitment to the organization declined**

The studies [17, 18] augment the need to create harmonious and cooperative teams. Such harmony is accomplished by focusing on the secondary Dynamics Coaching role of the PC; harnessing and resolving conflict. Conflict is generally thought of as bad and in most cases this is true, but in some cases, a specific type of conflict is very beneficial and can boost productivity, creativity, and harmony.

Relational Conflict is a term defined by Steven P. Robbins [19] that describes a mean-spirited and personal type of conflict involving differences between people and their personalities. These conflicts revolve around team members and behaviors and do not lend themselves to productivity or team harmony in any way. In fact Relational Conflict is the seedbed for mistrust.

The beneficial element of conflict was termed Functional Conflict [19] by Robbins and is described as two parties disagreeing about the functionality of an occurring problem or its functional solution. Some examples include two team members disagreeing about how to diagnose a problem, what methodology to use in developing a solution, disagreeing about the outcome of a procedure, or the amount of resources needed to accomplish a task. Personality is removed in each example and a focused discussion is held where both parties are focused on how to solve a problem rather than the people who are trying to solve a problem and therein is the difference. This thought process stems from a portion of our brains called the neocortex where rational and logical thoughts occur. It is where we perform reasoning, problem solving, decision making, impulse control, and limit the emotional portion of our brains. Try to be angry or sad while solving a complex math problem and you will find out it is next to impossible because the neocortex and the amygdala (the part of the brain that performs emotional reactions and stores emotional events) are separate in location and function; making it difficult for each to work in conjunction with one another. This mutually exclusive trait of the brain provides insight on how to bring relationally conflicted groups together using functional conflict. Functionally conflictive groups are proven to get closer to true resolution in a shorter amount of time than groups who do not practice conflict resolution type activities [20, 21, and 22]. Armed with the understanding of the dissonance of Relational and Functional Conflict, a PC can now be on constant lookout for burgeoning disputes and must be prepared to act upon them immediately by turning team focus from relationally conflictive topics to functionally conflictive ones.

Some conflicts are purely relational in nature and can be solved by more mature HPC techniques not discussed in this article, but many conflicts are functional in nature, but have been convoluted and complicated by relational conflict. The following steps are suggested to move a team from a relational to a functional state of conflict when functional issues are present:

1. **Gather the parties in dispute**
2. **Hold a discussion about the nature of the problem**
3. **Identify the functional elements of the problem**
4. **Lead and focus discussions on how to solve the functional aspects**
5. **Minimize and redirect relational comments**
6. **Hold follow-up sessions to discuss functional progress until progress is made**

Dr. Gerald Weinberg stated, "No matter how it looks at first, it's always a people problem [23]." Project teams have untapped resources and can solve innumerous problems when working together in harmony and building off each other's strengths. HPC's Dynamics Coaching element is designed to tap into those previously untapped resources to (1) shape and unite teams, (2) move projects and people toward success, (3) identify individual and team strengths, and (4) build trust and rapport without bringing in outside assistance that may slow or disrupt the sometimes fragile balance of project team dynamics. Dynamics Coaching should become a natural part of managing any project.

2. HPC and Workforce Development

HPC is also valuable in shaping a workforce for long-term success. The benefits to an organization include:

Strong Skills: Getting the right resources to the right people at the right time strengthens employees' technical and non-technical skills and improves the overall competence of the workforce.

Highly Competitive: HPC addresses humanistic factors that cause project failure, optimizes the skills and talents of individuals and teams, and mitigates the risks commonly associated with software engineering projects, which enables an organization to be much more effective and economical. This greatly increases competitiveness in the marketplace.

Agility: Because the strengths of employees and teams and the HPC principles are easily transportable to other projects, the workforce becomes much more adaptive and better able to respond to new, complex situations.

Productivity: HPC enhances the effectiveness of both individuals and teams through collaboration and synergy. This significantly increases productivity.

Employee Satisfaction: One of the greatest benefits of HPC is employee satisfaction. Empowerment, trust, and ownership are powerful motivators, and HPC builds trust between team members and management, values the individual, inspires the team, and motivates and rewards the workforce. This enables an

organization to attract and keep the best employees.

HPC plants seeds of excellence that can blossom anywhere, creating a highly competent workforce that thrives on change and challenges, and that is highly motivated and productive. This in turn makes the organization more competitive and able to accomplish diverse and profitable projects. HPC has a profound impact on the project, the workforce, and the bottom line.

Acknowledgements:

This paper is approved for public release; distribution is unlimited (reference No. 10-02-02_MXW_001). The authors would like to thank Stephen Shields of the Gallup Corporation for his assistance in providing data.

Disclaimer:

®CMMI is registered in the U.S. Patent and Trademark Office by Carnegie Mellon University.

ABOUT THE AUTHORS

Randall W. Jensen, Ph.D., is a Subject Matter Expert for the Software Technology Support Center, Hill Air Force Base, Utah, with more than 50 years of practical experience as a computer professional in hardware and software development. For the past 35 years, he has actively engaged in software engineering methods, tools, quality software management methods, software schedule and cost estimation, and management metrics. He retired as chief scientist of the Software Engineering Division of Hughes Aircraft Company's Ground Systems Group, and was responsible for research in software engineering methods and management. He founded Software Engineering, Inc., a software management-consulting firm in 1980. He developed the model that underlies the Sage and the Galorath Associates, Inc.'s SEER-Software Estimating Model [SEER-SEM] software cost and schedule estimating systems. Jensen received the International Society of Parametric Analysts Freiman Award for Outstanding Contributions to Parametric Estimating in 1984. He has published several computer-related texts, including "Software Engineering," and numerous software and hardware analysis papers. He has a Bachelor of Science in Electrical Engineering, a Master of Science in Electrical Engineering, and a doctorate in Electrical Engineering from Utah State University.

Randall W. Jensen, Ph.D.
Software Technology Support Center
6022 Fir AVE, Bldg. 1238
Hill AFB, UT 84056-5820
Phone: (801) 775-5742
Fax: (801) 777-8069
E-mail: randall.jensen@hill.af.mil

Joyce Peters has worked as an Air Force civilian since 1985. With experience in every area of civilian personnel, and also in acquisition and Air Force intelligence, she is now responsible for strategic planning and shaping of the civilian workforce at Hill AFB. She has worked closely with the Air Force on regulations and policies to insert performance requirements analysis, HPT and learning on demand into Air Force education and training processes. She is also liaison between Hill AFB and the state of Utah on educational partnerships and workforce development issues.

Kasey Thompson is an advisor to CrossTalk Magazine and leads the Software Organizational Development Office at Hill AFB where he consults, coaches, and instructs in the areas of project management, emotional intelligence, and teaming issues. He also owns and operates Human Investments, a consulting firm aimed at improving relationships within corporations, families, and married couples. He received a BS in Lifestyle Management, an MBA, and is an Arbinger Institute trained coach.

Dr. Doretta E. Gordon is Director of the Emerging Training and Performance Technologies Department at Southwest Research Institute. She has over a decade of research and experience in the fields of Instructional Systems Design and HPT. Dr. Gordon has served as a visiting instructor at the university level, teaching courses in performance analysis, instructional systems design, and technology infusion. Her research interests include ubiquitous learning and the elicitation and transformation of expert tacit knowledge.

Fred Smullin is Business and Technology Manager for Woodbury Technologies, Inc. He is leading a companywide initiative to map all business and KM processes as well as work with departments to recommend improvements using standardized documentation and simulation techniques. He achieved his ITIL version 3 foundation certification and is participating in company CMMI® Level 2 certification efforts. He has consulted and developed software over the past 20 years for DoD, industry, and international clients.

REFERENCES

1. Bauer, F.L. (Chair), Bolliet, L. and Helms, H. J. (Co-Chair). Software Engineering. Report on a Conference Sponsored by the NATO Science Committee. Garmisch, Germany. 1968.
2. Charette, R. N. Why Software Fails, IEEE Spectrum. September 2005.
3. McGregor, Douglas, The Human Side of Enterprise (New York: McGraw-Hill Book Company), 1960.
4. Deming, W. Edwards, Out of the Crisis (Cambridge, Mass: MIT Press), 1982.
5. Boehm, B. Software Engineering Economics. (Englewood Cliffs, NJ: Prentice-Hall), 1981.
6. SEER for Software, <http://www.galorath.com>.
7. Dwyer, Rocky J. (2009) Prepare for the impact of the multi-generational workforce!, Transforming Government People, Process and Policy, Volume 3, Number 2, pp 101-110.
8. Bellinger, G. , Castro, D. and Mills, A. Data, Information, Knowledge, and Wisdom, <http://www.systems-thinking.org>
9. Gilbert, T. F. (1978). Human competence: Engineering worthy performance. (New York: McGraw-Hill), 1978.
10. Rummler, G. A., & Brache, A. P. (1990). Improving performance: How to manage the white space on the organization chart. (San Francisco: Jossey-Bass), p. 13.
11. Gilburg, Deborah, 2007, Management Techniques for Bringing out the Best in Generation Y, CIO Magazine, <http://www.cio.com>
12. Gilburg, Deborah, 2007, Generation X: Stepping Up to the Leadership Plate, CIO Magazine, <http://www.cio.com>
13. Web 2.0, <http://en.wikipedia.org>.
14. Rath, T., & Conchie, B. (2008). Strengths Based Leadership: Great Teams, Leaders, and Why People Follow. (New York: Gallup Press), p 12.
15. Gallup Management Journal News Releases (Feb. 12, 2007). New Book Continues the Strengths Revolution, Gallup Management Journal <http://gmj.gallup.com/content/26512/New-Book-Continues-Strengths-Revolution.aspx>
16. Covey, Stephen M.R., The speed of Trust: The One Thing that Changes Everything, (New York, NY, Free Press), 2006.
17. Costa, Ana C., Roe, Robert A., &Taillieu, Tharsi. Trust within teams: The Relation with Performance Effectiveness, (European Journal of Work and Organizational Psychology). 2001 p
18. Pearson, Christine, & Porath, Christine, The Cost of Bad Behaviors – How Incivility Is Damaging Your Business and What to Do About It, Portfolio (Penguin Group USA), (New York, NY), 2009.
19. S.P. Robbins, The Truth About Managing People: And Nothing But the Truth, (Upper Saddle River, NJ: Prentice Hall), 2003.
20. K.A. Jehn, "A Qualitative Analysis of Conflict Types and Dimensions in Organizational Groups," Administrative Science Quarterly, September 1997, pp. 530-57.
21. C.J. Nemeth, J.B. Connell, J.D. Rogers, and K.S. Brown, "Improving Decision Making by Means of Dissent," Journal of Applied Social Psychology, January 2001, pp. 48-58.
22. K.A. Jehn, and E.A. Mannix, "The Dynamic Nature of Conflict: A Longitudinal Study of Intragroup Conflict and Group Performance," Academy of Management Journal, April 2001, pp. 238-51.
23. G. M. Weinberg, The Secrets of Consulting: A Guide to Giving & Getting Advice Successfully, (New York, NY, Dorset House), 1985.

An Agile Systems Engineering Process
The Missing Link?

Matthew R. Kennedy, DAU
David A. Umphress, Ph.D., Auburn University

Abstract: Today's systems are increasingly threatened by unanticipated change arising from volatility in user requirements, Information Technology (IT) refresh rates, and responses to security vulnerabilities. With the rapidly changing world of IT, long static development cycles of a Software Intensive System (SIS), a system in which software represents the largest segment in one or more of the following criteria: system development cost, system development risk, system functionality, or development time [1] may doom the system before development begins.

Delivering a SIS that is on time, within budget, and on schedule has been shown to be problematic [2]. This problem will only increase as the complexity of SISs within the DoD grows and more functionality within systems is relegated to software [3] [4].

Traditional systems engineering portrays systems development as a top-down, waterfall-centric process, one that relies on explicating requirements as early as possible. Such a perspective tends to postpone modifications until the maintenance phase [5], thus thwarting early insertion of technology or a nimble response to changes in user needs. Though the technology refresh rate varies from system to system, a report from the state of Michigan shows the following industry computer technology refresh trends:

1. 40% of companies are on a four-year cycle for refreshing personal computers (hardware), and
2. Microsoft plans a two-year cycle to release a new operating system (software) [6].

A report from the U.S. Army War College estimates that commercial electronics have a typical refresh rate of 12-18 months but may be less [7].

Cyber security further complicates the picture. The rate at which vulnerabilities are identified in a system cannot be predicted. According to the National Vulnerabilities Database, between 2000 and 2009 there was an average of 3,825 vulnerabilities reported each year due to software flaws alone [8]. The need for a responsive systems engineering process to rapidly address unforeseen vulnerabilities is imperative for the development of a secure system.

The Office of the Assistant Secretary of Defense for Network Information Integration conducted an analysis of 32 major information system acquisitions and found the average time to deliver the Initial Operating Capability was 91 months [3]. With the DoD's history of long delivery cycles and the short time required for technology refresh, the systems engineering process needs to be responsive to changes introduced both by the user and technology.

This inability to respond rapidly to change is nothing new. Software engineering recognized the pitfalls of a strictly sequential development process a number of years ago. The contemporary school of thought in software engineering has evolved away from considering a waterfall approach as the primary sequence of development activities and toward approaches that embrace change by segmenting software development into manageable change-resistant increments and allowing change to take place at increment boundaries [5]. Ultra-modern approaches–known as agile processes–have emerged to match the pace in which change is encountered during software development. Agility is "the speed of operations within an organization and speed in responding to customers (reduced cycle times)" (Massachusetts Institute of Technology). The degree of agility when developing an IT system is the organization's ability to respond to changing requirements and technology. With the quick technology refresh rate, long development cycles could place a system in a state of obsolescence prior to initial release. With the ever-changing world of technology, the need to change without notice throughout the development lifecycle is paramount to success.

Just as the software community has moved toward a more agile approach to become more responsive to changes throughout the development lifecycle, the systems engineering community needs to follow a similar approach to remain competitive in today's rapidly changing environment.

Past Performance

Failure to deliver a successful SIS can rarely be attributed to one project deficiency; however, the inability to rapidly adapt to change appears to be an underlying theme in many SIS development failures. A successful SIS is defined as a system that is on time, within budget, and contains all of the required features and functions [9]. Instead of steadily making improvements on the successful delivery of SISs, the Standish Group 2009 Chaos report showed a "marked decrease in project success rates," in

which only 32% of projects were successfully delivered when compared to the 35% reported in their 2006 report [9] [10].

The U.S. Government Accountability Office (GAO), an "independent, nonpartisan agency that works for Congress," investigates how the government spends taxpayers' dollars [11]. The Air Force is developing an F-22 aircraft that is intended to provide increased capabilities over current aircraft. A GAO report found the program has undergone several changes since the development began in 1986 and the Air Force cannot afford to purchase the quantities of the aircraft that were initially anticipated. This was partially attributed to the Air Force adding more robust air-to-ground attack requirements in 2002. In addition to the change in requirements, the Air Force has determined that a revised computing architecture, as well as new computer processors were needed to support planned enhancements, both of which further increased program costs [12]. Previous experience shows that changes within a SIS are inevitable, whether or not there is a change in requirements or technology. Though predicting these changes may be difficult, processes can be structured to be more responsive to these unanticipated changes. Increasing agility within the systems engineering process is one mechanism that may result in increasing the successful delivery of a SIS.

Growth of SISs

The software within today's systems is only increasing. Examining the correlation between the Executable Software Lines of Code (ESLOC) and time in various DoD systems (Figure 1) shows a steady increase in ESLOC in related systems over time. The Aegis system introduced in the early 1980s had less than 2 million ESLOC. The Virginia SSN introduced roughly 20 years later contained over double the ESLOC and the estimated ESLOC for the DDX system is just under 10 million.

The increase in ESLOC means that more of the system's functionality is being performed by software. Functions performed by software in DoD aircraft (Figure 2) has increased from 8% for the F-4 Phantom II in 1960 to 80% for the F-22 Raptor in 2000. With the proliferation of software within current systems, problems that were inherently software are evolving into system problems [4].

DoD systems are not the only systems experiencing an increase in software; the automotive industry has also seen an increase. In 1977 the Oldsmobile Toronado contained the first productive microcomputer Electronic Control Unit used for only electronic spark timing [13]. Just a year later, the Cadillac Seville offered on its *Cadillac Trip Computer* a software-driven display of speed, fuel, trip, and engine information [13]. By 1981, GM was using microprocessor-based engine controls executing roughly 50,000 Software Lines of Code (SLOC); today it is estimated that a premium automobile takes dozens of microprocessors running 100 million SLOC [13].

When determining the impact of software on overall system cost, Broy notes that "the cost of software and electronics can reach 35% to 40% of the cost of a car [13]. A study conducted by the Center for Automotive Research had similar findings [14] stating, "Software made up only 16% of a vehicle's total value in 1990, this figure had increased to 25% by 2001. By 2010, the share of a car's total value is expected to climb to almost 40%."

The inability to deliver a successful SIS will only be exacerbated as software continues to become an increased portion of a system's composition.

Figure 1: Increase in Software in DoD Systems

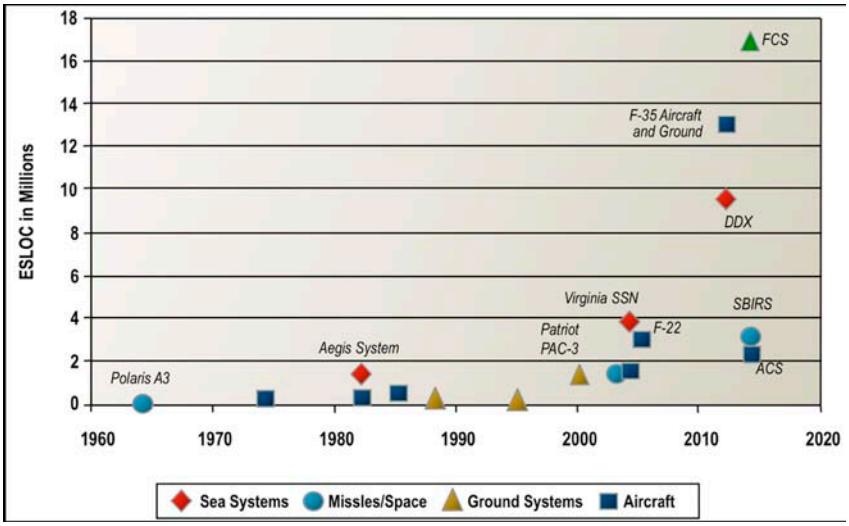
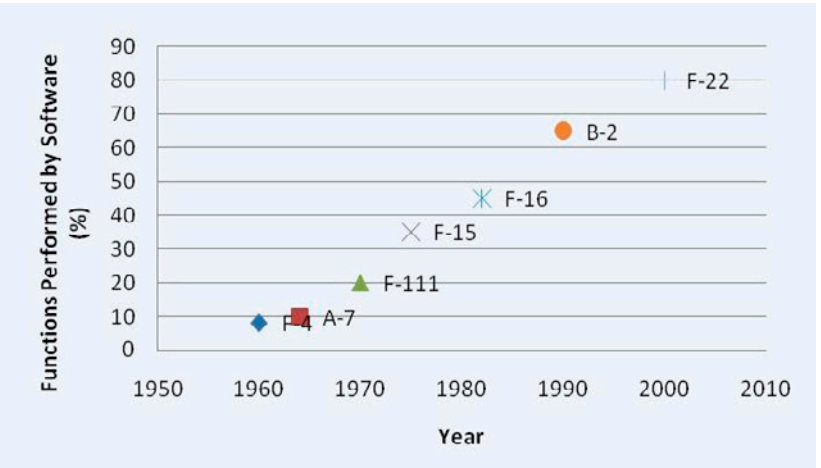


Figure 2: Functions Performed by Software (Nelson and Clark)



SIS Development

Development of a SIS can be envisioned as an amalgamation of three aspects: business, system, and software. Though there is some overlap among these aspects, general responsibilities can be attributed to each aspect.

The business aspect is responsible for the overall acquisition of the system including contracting, funding, operational requirements, and overall system delivery structure. The system aspect is responsible for the overall technical and technical management aspects of the system and serves as the interface

Figure 3: Std 1220-2005 Systems Engineering Process

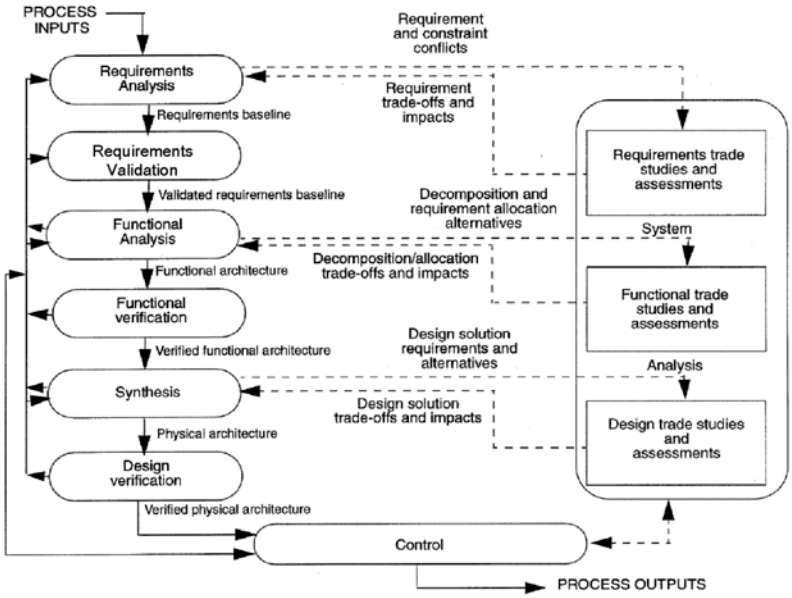
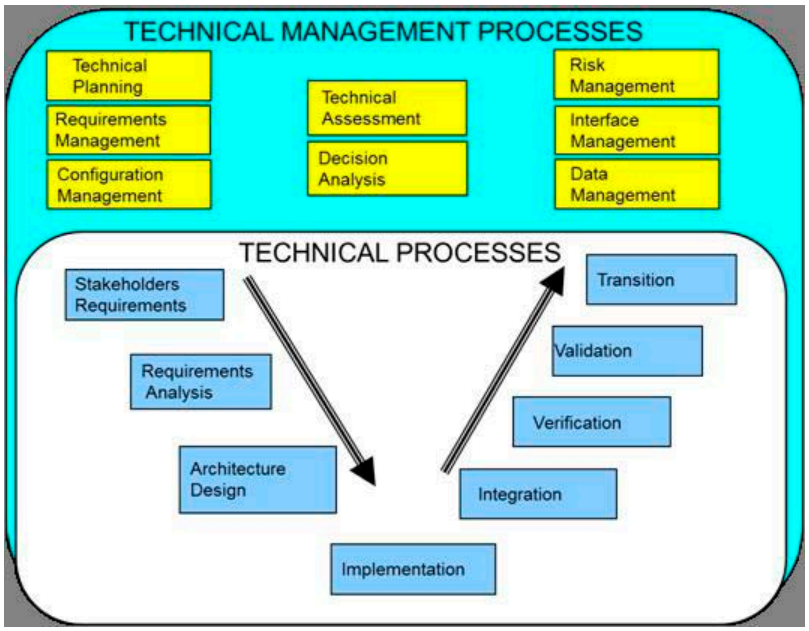


Figure 4: DAG Systems Engineering Processes (University, DAU Information Resource Management 202 Course)



between management and engineers. The software aspect is responsible for the software items contained in the SIS.

When developing a SIS, all three aspects need to work in harmony to produce a successful final product. Traditionally, when using a once-through development methodology, the business aspect would provide the funding and operational requirements to the system aspect. The system aspect would further decompose the requirements and allocate them to software or hardware. These items would then be developed and integrated resulting in a completed system. Given that major information systems average a 91-month gap from operational requirements definition to system delivery, defining requirements that far in advance of technology that is changing every 12 to 18 months suggests that the end result will not be an up-to-date system.

The need for increased agility has been identified within the business aspect and there are initiatives aimed at developing an agile framework within this aspect. Per the fiscal year 2010 National Defense Authorization Act, section 804, the U.S. Congress directed the Secretary of Defense to, "develop and implement a new acquisition process for IT systems" [15]. This new Defense Acquisition System process must include: Early and continual involvement of the user; multiple, rapidly executed increments or releases of capability; early, successive prototyping to support an evolutionary approach; and a modular, open-systems approach [15].

Moreover, this process should be based on the March 2009 report of the Defense Science Board (DSB) Task Force on Department of Defense Policies and Procedures for the Acquisition of Information Technology [15]. The DSB report concluded, "The conventional DOD acquisition process is too long and too cumbersome to fit the needs of the many IT systems that require continuous changes and upgrades" [3].

The report noted that an agile acquisition approach would increase IT capability and program predictability, reduce cost, and decrease cycle time.

In addition to the emerging Agile IT Acquisition Lifecycle, the DoD developed an agile requirements process for IT Systems called the "IT Box" [16]. The Joint Requirements Oversight Council Memorandum 008-08 stated, "IT programs are dynamic in nature and have, on average, produced improvements in performance every 12-18 months" [17]. Recognizing the need for performance improvements, the IT Box allows IT programs the flexibility to incorporate evolving technologies.

Lack of evidence implies the system aspect does not have similar agile initiatives. There are several systems engineering guides and standards available such as the Defense Acquisition Guidebook (DAG) Chapter 4, EIA-632, IEEE std 1220-2005, ISO/IEC 15288, and ISO/EIC 26702 [18,19,20,21,22]. In practice, no single systems engineering standard is used, but instead a combination of standards. For example, the Air Force produced Instruction 63-1201, Life Cycle Systems Engineering, which references numerous systems engineering standards and is to be used in the development of all AF systems [23]. These guides and standards provide the overall structure of the systems engineering process as well as identify characteristics required during the process.

IEEE Std 1220-2005 defines a systems engineering process (Figure 3) as, "a generic problem-solving process, which provides the mechanisms for identifying and evolving the product and process definitions of a system." It further notes that the SEP should be applied throughout the system lifecycle for development and further identifies the lifecycle stages (System definition stage, Preliminary design stage, Detailed design stage, Fabrication, assembly, integration, and test stage, Production and customer support stages). However, it does not detail how the SEP should be applied from an agile project management perspective.

In contrast to IEEE Std 1220-2005, the DAG, Chapter 4, divides the SEP into two categories: Technical Management Processes and Technical Processes [18]. At a high level, the generic Technical Processes frame the steps necessary to develop a system whereas the Technical Management Processes are used to manage the technical development (Figure 4).

In addition to further describing key activities in each process area, the DAG contains some systems engineering best practices such as employing a modular design and designating key interfaces [18].

Current systems engineering guides and standards provide a waterfall-like structure and key systems engineering characteristics that are imperative for successful system development. However, they do not provide a framework for planning and managing projects that allow systems engineers to rapidly respond to the changes. The design and implementation of such a framework is left to the systems engineers who are provided little guidance. The structure and characteristics provided need to remain intact while their application needs to be framed such that it allows for an agile implementation.

Similar to the system aspect, the software aspect has a number of standards available such as ISO 12207, ISO 9001 and the Capability Maturity Model Integrated (CMMI®) [24,25,26]. The CMMI was a collaborative effort by the U.S. government, industry and Carnegie Mellon [27] that contains a process improvement model consisting of best practices addressing activities throughout the products lifecycle [24].

ISO 12207 "contains processes, activities and tasks that are to be applied during the acquisition of a system that contains software" [26]. A limitation identified within ISO 12207 is that it does not specify details on how to implement the identified activities or tasks [26].

As with the system aspect, the software aspect guides and standards only provide the characteristics required; however, the software aspect has agile frameworks built on top of these standards, that allow software to be developed in an atmosphere where requirements are changing. One such agile framework is called Scrum. Scrum was formalized by Ken Schwaber at the Object-Oriented Programming, Systems, Languages and Applications conference in 1995 [28]. Since Scrum has been in existence for 15 years, it has a large collection of lessons learned, as well as success stories, which have contributed to its current state. These additional frameworks allow the Software Aspect increased agility during the development process.

Conclusion

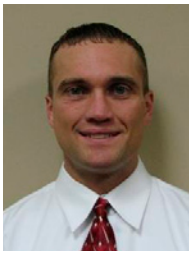
The rapid technology refresh rate coupled with the need to respond to changing requirements requires a complete agile development process; one where the business, system, and software areas contain an agile framework and work in unison to create a successful SIS. A deficit in any of the three areas will cripple the overall process. The increase in software within today's systems only increases the need for an agile systems engineering process.

The emerging DoD Agile IT acquisition lifecycle and IT Box provide the foundation for the business area's transformation to agility. Currently, nothing is being done to address the lack of responsiveness within the system area. The system area provides the critical link between the business and software areas; as such, lack of agility in the system area can have a debilitating effect on the overall development process. This increases the risk of negating both the improvements being made in the business area and the existing agile processes in the software area.

The development of an agile system engineering framework is required to enhance the overall effectiveness of the SIS development process. Key interfaces also need to be identified from the system area to the business and software areas enabling seamless communication between adjacent areas.

Disclaimer:

ABOUT THE AUTHOR



Matthew R. Kennedy is a professor of software engineering at DAU. He served in the U.S. Air Force as a network intelligence analyst and he has more than 10 years of experience in IT. He has a bachelor's and master's degree in computer science.

David A. Umphress, Ph.D., is an associate professor of computer science and software engineering at Auburn University, where he specializes in software development processes. He has worked over the past 30 years in various software and system engineering capacities in military, industry, and academia settings. He is an IEEE certified software development professional.

REFERENCES

1. Defense Acquisition University. "Glossary." 2009.
2. The Standish Group. CHAOS Summary 2009. Boston, 2009.
3. Force, Defense Science Board Task. Department of Defense Policies and Procedures for the Acquisition of Information Technology. Washington: Office of the Under Secretary of Defense, 2009.
4. Ferguson, Jack. "Crouching Dragon, Hidden Software: Software in DoD Weapon Systems." IEEE Software (2001): 105-107.
5. FORCE, DEPARTMENT OF THE AIR. "Guidelines for Successful Acquisition and Management of Software-Intensive Systems." 2003.
6. Information Technology Equipment Life-cycle. Michigan, 2004.
7. Daniels, Jody. Review of Acquisition for Transformation, Modernization, and Recapitalization. Carlisle: U.S. Army War College,Carlisle Barracks, 2006.
8. Statistics. 23 10 2010. 23 10 2010 <http://web.nvd.nist.gov/view/vuln/statistics-results?cid=4>.
9. The Standish Group. CHAOS Summary 2009. Boston, 2009.
10. Dominguez, Jorge. "The Curious Case of the CHAOS report 2009." 2009.
11. About GAO. 11 09 2010 <http://www.gao.gov/about/index.html>.
12. Office, United States General Accounting. "Changing Conditions Drive Need for New F/A-22 Business Case." 2004.
13. Charette , Robert N. "This Car Runs on Code." IEEE Spectrum 2009.
14. "Electronics: Driving Automotive Innovation." Pictures of the Future 2005, Fall ed.
15. America, One Hundred Eleventh Congress of the United States of. "National Defense Authorization Act for Fiscal Year 2010." 2010.
16. Wells, Charles (LTC). "Information Technology Requirements Oversight and Managment (The "IT Box"" 2009.
17. JROC. "Joint Requirements Oversight Council." 2009.
18. "Defense Acquisition Guidebook." 2010.
19. ISO/IEC. "Systems and software engineering - System life cycle processes." 2008.
20. "Systems engineering – Application and management of the systems engineering process." 2005.
21. IEEE. "IEEE Standard for Application and Management of the Systems Engineering Process." 2005.
22. ANSI/EIA. "Processes for Engineering a System." 1999.
23. Force, Secretary Of The Air. Life Cycle Systems Engineering. 2007
24. CMMI® for Development, Version 1.2. Pittsburgh: Carnegie Mellon University, 2006.
25. "ISO 9001." Quality Management Systems. 2008.
26. Standardization, International Organization for. "ISO 12207." Software Life Cycle Processes. 2008.
27. Software Engineering Institute - Carnegie Mellon. "Brief History of CMMI." n.d.
28. Sutherland, Jeff and Ken Schwaber. "The Scrum Papers: Nut, Bolts, and Origins of an Agile Framework." 2010.

From MBWA to LBWA

21st Century People Solutions for Software Problems

Jonathan Powell, CACI

Abstract: People solutions to software problems is a misnomer. In the final analysis, people are the only solution to software problems. A qualified team built on trust and engagement optimizes organizational productivity and can solve any issue with software.

Introduction

If the DoD is to remain the wellspring of U.S. technological innovation in the 21st century, it must develop and institutionalize new and different approaches to people management.

The changes required run the gamut, from recruiting to day-to-day working conditions. Some of these areas have started to be addressed widely (examples include streamlining the recruiting process and expanding telecommuting). This article includes advice on some of the areas already examined, as well as an area that hasn't received as much attention, but is as vital (if not more so) to maximizing the organization's productivity and opportunities for innovation.

We have not evolved to the point where software is capable of designing, building and deploying itself. Nor have we reached the point where, once deployed, the software can configure, maintain, and enhance itself. Sure, certain advancements have been made. Automation is now found throughout the software development lifecycle, from requirements through operations and maintenance. This includes requirements gathering and modeling software through "self healing" systems and artificial intelligence.

However, as much as software and software development have evolved, it still comes down to people who design and build the software, and then intercede and fix the software when it does not work or behave the way it is supposed to.

Today when the Navy needs next generation software for its submarine sonar systems, defense contractors are not deploying hordes of automatons to the Pentagon to gather requirements, design the software, build the prototype, make it ready for production, and then support it through operations and maintenance. It still comes down to people, and I submit it will come down to people for a long time to come. Even in the far out future, as parts of this chain are automated, people will be needed to intercede, because software is not perfect, and problems always arise.

So if one's ability to overcome software challenges fundamentally comes down to people, the question becomes, "How do I get the most out of my people?" This will be the focus of this article.

Hiring

Since people are the key to overcoming software challenges, this essential topic needs to be addressed on the front end. Everyone understands the basics of hiring–what skill sets do I need and for what level of personnel (junior, mid-level, senior). What so many teams fail to do is institute rigor with respect to a person's "fit." All corporations have a culture, and within that umbrella one can find variations on different software development teams. As program manager for a large, complex, custom software development effort for the U.S. Army, I have personally interviewed all developers brought on to the team. The program peaked at 60 people, so maintaining this policy was certainly challenging. However, I viewed it as vital, and I'm not alone. For decades Admiral Rickover personally interviewed all Junior Officers who entered the nuclear Navy. Jim Collins in the book *Good to Great* cautioned on the need to be slow to hire. Robert Townsend, former CEO of Avis wrote, "The important thing about hiring is the chemistry or vibrations between boss and candidate: good, bad, or not there at all" [1].

Fit is important because it helps minimize transition costs. Hiring is costly. These costs are exacerbated if someone is brought in who requires extra training, or doesn't follow the team's behavioral norms, therefore negatively impacting productivity. Worst case, the person cannot adjust and has to be removed.

Best case, the person is able to adjust to function seamlessly within the team. And this best case is exceedingly rare, because for experienced professionals, change does not come easily as one's work patterns have become set over a number of years.

Let me give you an example. We had a developer who did not fit. One of his major issues was not being able to collaborate effectively with his fellow developers, which severely impeded his progress because it is a complex object oriented system. Since he failed to collaborate with his peers, his code was frequently rejected by the Test Team. Here he failed in another major way–instead of working with the Test Team to understand their reasons for rejection and establishing a way forward, he would simply adjust the software to his liking, and toss it back over the wall to Test, where it was invariably rejected again. By the time the issues with this developer were escalated to my level, it was clear he was set in his behavior and was not going to change. We put him on a performance improvement plan in accordance with company policy and had to dismiss him when he did not improve as the plan stipulated.

We have implemented some best practices to help avoid this situation. First is utilizing an interview panel, to include multiple developers. A variety of perspectives are important to help ensure we're bringing on someone with the right fit. Second, we include members from multiple teams where possible. So, for a developer interview, we would include interviewers not just from the team he or she is interviewing for, but also a developer from another development team on the program. If possible, we would also include a tester in the interview process. Communication and collaboration intra-team and across teams is vital for our program's success, and we want to bring in people who can function well in this environment.

This process is not perfect and certainly is more costly, so the reader needs to weigh the costs-benefits as it pertains to his or her program, but for large complex programs like ours, the up-front investment is more than outweighed by the costs and risks introduced by bringing in the wrong type of person. If you take shortcuts with your hiring, and do not ensure as a first step you are getting people who will fit in with your group and productively work with the team you have in place, you will fundamentally undercut your ability to respond to the challenges you will invariably encounter during the software development lifecycle.

Work Environment

If people are the solution to software problems, and we have selected candidates who we believe can contribute to the program's success and succeed, the question then becomes how do we get the most from our people? Here the answer starts and ends with environment. How your team performs today and responds to the myriad software development challenges that emerge along the way all depends on the work climate you establish and foster in your organization. "Provide the climate and proper nourishment and let the people grow themselves. They'll amaze you" [2].

Ask yourself some questions. Is everyone made to perform the same way in your program? Or are roles tailored to the particular strengths of each person? Are you fitting people into your organization or are you building team cohesion around the individual strengths of its members? Townsend wrote, "Why spend all that money and time on selection of people when the people you have got are breaking down from underuse. Get to know your people–what they do well, what they enjoy doing, what their weaknesses and strengths are, and what they want and need to get from their job. Then try to create an organization around your people, not jam your people into those organization-chart rectangles … You cannot motivate people. That door is locked from the inside. You *can* create a climate in which most of your people will motivate themselves to help the company reach its objectives" [3].

What Silicon Valley has and continues to accomplish is a testament to the soundness of this thinking. In the '80s and '90s it achieved notoriety for its unconventional work practices–casual clothing, foosball tables and other amenities, flextime according to personal needs, etc. Of course this notoriety was eclipsed by the operating results and history-changing evolutions that came from the Valley, innovation that continues apace to this day.

You might be thinking, "All of that sounds great Jonathan, but how do I make this work in the world of DoD software development?" The answer is remarkably simple and it starts with walking around. That is right–getting out of your office or cubicle and personally interacting with your developers, engineers, analysts, and other team members. While, as Watts S. Humphrey stated, one can no longer use Management By Walking Around for knowledge work, he or she can and must use what I call Leadership By Walking Around (LBWA).

LBWA and Other Keys to Success

LBWA is the perfect way to get to know your people and observe how they are functioning. It also shows as a manager you are interested and care, which will inspire and motivate those around you, as well as foster an environment of open communications, critical to any project's success. Walking around is one of the simplest, yet most vital tools in providing effective leadership, not just because of all the things you learn about your teammates and the open communications you inspire, but most important of all, because of the example you set. J. Paul Getty put it best when he wrote, "No psychological weapon is more potent than example. An executive who seeks to achieve results through the people who work under his direction must himself demonstrate at least as high a standard of performance as he hopes to get from his subordinates" [4].

Furthermore, LBWA helps build trust, and trust is the oil that makes an organization truly hum as a well-oiled machine. "Trust is what motivates people to follow our leadership, whether at work or home. And trust must be earned" [5]. And what better way to earn trust than by visiting people in their space and truly listening and interacting with them on a personal level? If it works for world leaders and diplomats for affairs of state, cer-

tainly it should work at the project level with day-to-day software matters. The trust you will build will help unlock individual motivation for the mission and increase engagement by the team. And as the CEO and President for the Partnership for Public Service stated "Effective Leadership is the No. 1 most important issue for employee engagement."

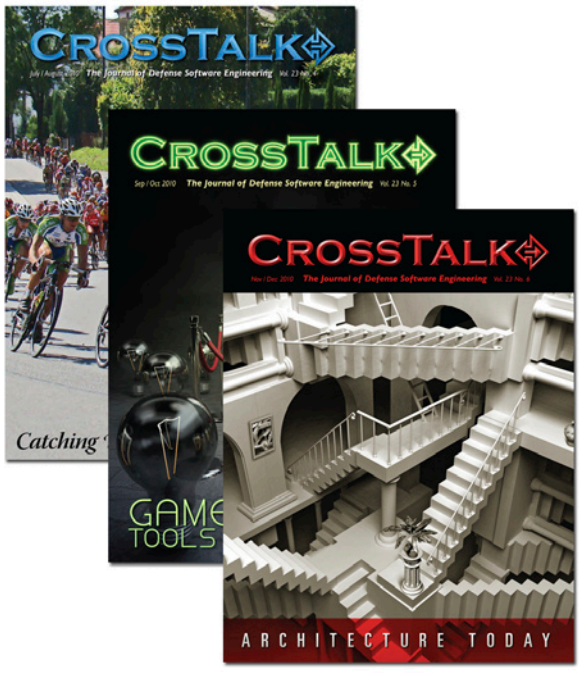
LBWA is not the only tool in your toolkit. However, as the leader, it is incumbent upon you to start with this fundamental–not aimless wandering around mind you, but prudent, targeted engagement.

Beyond LBWA, there are a number of widely known tools that will help improve one's ability to lead effectively. These include open-door management policies, conducting all hands meetings, e-mail messages directly from senior leaders, brown bag lunches, and hosting off-site sessions. These items are all very much relevant and useful within the DoD software development context, even more so because these are cost effective and simple to employ, and well within established military cultural norms. Recently I've successfully employed all of these tactics with a large Army customer comprised of many varied and dispersed stakeholders. Moreover, I've been able to leverage the military's Integrated Program Team framework to increase the effectiveness of these tools. For example, on the eve of entering a particularly intense high operational tempo period, a senior Army proponent was kind enough to help me lead an all hands session where we jointly presented to the 55-person contractor team. He and his staff have also written e-mail messages of commendation to program team members. These sorts of efforts tend to have an exponential reinforcing impact on morale and engagement when appropriately used to supplement company-internal communications.

Other ways exist to further leadership effectiveness and thus increase employee engagement, which in turn increases a program's ability to tackle and overcome software challenges. Some of these methods may require more creativity than others, but they are all still relevant and readily applicable within the DoD space. For example, tailoring work hours to the time of day when an employee is personally most effective. It is widely accepted that people have different biorhythms and perform better and are therefore more productive during certain parts of the day. Your program should try to support these individual needs to the extent practicable. This notion is a lot more accepted today in the DoD than it used to be. Work practices have evolved to include considering how we cater to the differing needs of three workforce generations–Baby Boomers and Generations X and Y, respectively. In particular, when it comes to attracting and retaining younger talent, implementing more progressive policies than those seen in decades past are imperative if the DoD is going to thrive in the 21st century. Certainly DoD cannot hope to be the wellspring of creativity in this era if the environment is a disincentive, especially with young men and women now having career options to choose from around the world.

Increasingly the DoD has moved away from contractually mandated work hours to performance-based work. Still, corporations catering to this sector too frequently default to standard working hours instead of examining how they can best maximize the productivity of each individual in a way that optimizes the whole organization. Same thing with work location, and the increasing traction telecommuting is achieving. Historical boxes and walls of work location, hours, etc. are being shunned in favor of management theory seeking to provide the employee with the best fit possible, instead of vice versa, thus enabling him or her to provide the organization with his or her best work.

The DoD is a large organization you say, so what can I do then if I work in a part of this sector where flexibility around work practices has not caught on? I would encourage you to look again. Even within the strictest confines of traditional DoD, I believe as a leader you have both an opportunity and an obligation to be creative for your people. Take a look–you will be surprised by what you find.



CALL FOR ARTICLES

If your experience or research has produced information that could be useful to others, CrossTalk can get the word out. We are specifically looking for articles on software-related topics to supplement upcoming theme issues. Below is the submittal schedule for three areas of emphasis we are looking for:

Software's Greatest Hits and Misses
November/December 2011
Submission Deadline: June 10, 2011

High Maturity - The Payoff
January/February 2012
Submission Deadline: Aug 10, 2011

Securing a Wireless World
March/April 2012
Submission Deadline: Oct 10, 2011

Please follow the Author Guidelines for CrossTalk, available on the Internet at <www.crosstalkonline.org/submission-guidelines>. We accept article submissions on software-related topics at any time, along with Letters to the Editor and BackTalk. To see a list of themes for upcoming issues or to learn more about the types of articles we're looking for visit <www.crosstalkonline.org/theme-calendar>.

For example, many services-based contracts adhere to Labor Categories (LABCATs), including formal job descriptions. Robert Townsend called job descriptions, "Strait jackets … insane for jobs that pay $750 a week or more. Judgment jobs are constantly changing in nature and the good people should be allowed to use their jobs and see how good they are" [6].

Practically, we all understand this, as folks are constantly asked to assume expanded roles and/or assume collateral duties. Since we cannot completely rely on a job description as a gauge, as a first step, make sure you have your folks playing the right positions and if they are in the right position, ensure they are in the correct role. This means doing work beyond a crosswalk of a person's skills/experience/education to LABCAT requirements, but an in-depth qualitative check for real organizational fit. For example, some developers prefer bug fixes over operations and maintenance.

Others prefer building new capabilities. And you can take this analysis down to finer levels of detail, to ensure a fit is made that resonates and is optimal for the individual and organization.

How about attire? The Army program I have cited permits personnel working outside of the Pentagon to dress "business comfortable." That does not mean folks are running around in tank tops and flip flops. In fact, there has not been a single incident of someone going over the line. If the mindset is to treat individuals as adults and with trust, the employees are more engaged, and you also increase your odds of attracting the best talent.

Unleashing the Organization

A good work environment fitted to support individuals and not the other way around, coupled with effective leadership, leads to engaged employees. And engaged employees are the key to unleashing the power of your team to solve problems.

Robert Zawaki observed in *Transforming the Mature Information Technology Organization* the following:

ED = RD X CD [7]

An Effective Decision (ED) is equal to the Right Decision (RD) multiplied by the Commitment to the Decision (CD). Employee engagement is central to both variables in the equation. Developing the right decision will require team involvement. Only by leveraging their collective wisdom, across functions, will you arrive at the RD. While some projects get this aspect right, often overlooked is how vital CD is to success. By involving the team in deriving RD, much of the heavy lifting in increasing CD is done. In executing a transparent collaborative decision-making process, you will be far ahead in your ability to get team members to commit to the best course of action, the RD. This is because they will have seen the logic used to arrive at the RD and how various perspectives were raised and used to shape the RD, including their own. Since both RD and CD are needed for a software solution to succeed, the effective leader will not omit the time and attention needed to maximize these.

Summary

In today's hyper competitive global environment, all organizations are focused on maximizing productivity. What is too often overlooked is the best way to achieve this. Bottom-up is the way to go–determining how the organization can be molded to unleash productivity on an individual basis, while maintaining a system to optimize the whole. Effective leadership is the key to shaping the organization to maximize employee productivity, because it spurs employee engagement. By maximizing the output of the entire team, and harnessing it for mission accomplishment, the effective leader can overcome any software problem through people. "The capacity of people to find answers, if they know it is worth the trouble, has never been tested to its practical limits" [8].

ABOUT THE AUTHOR

Jonathan W. Powell, CGFM, PMP is a Senior Program Manager with CACI. A former submarine officer, Mr. Powell has led complex engagements for military, federal, and intelligence agencies. His articles have been published in PM Network and Contract Management, respectively. Mr. Powell serves on the Board of the Montgomery County Revenue Authority, where he is a member of its finance committee. He holds a B.S. from USNA and an MBA from the University of Maryland.

Jonathan Powell
CACI
14370 Newbrook Drive
Chantilly, VA 20151
Phone: 703.258.4735
E-mail: jopowell@caci.com

REFERENCES

1. Townsend, Robert. Up the Organization. San Francisco, CA: Jossey-Bass, 2007. Page 96.
2. Townsend, Robert. Up the Organization. San Francisco, CA: Jossey-Bass, 2007. Page 96.
3. Townsend, Robert. Up the Organization. San Francisco, CA: Jossey-Bass, 2007. Page 95.
4. Getty, J. Paul. How to be Rich. New York, NY: Penguin Group, 1965. Page 80.
5. Farrar, Steve. Point Man. Frisco, TX: Multnomah Books, 1990. Page 153
6. Townsend, Robert. Up the Organization. San Francisco, CA: Jossey-Bass, 2007. Page 59
7. See <http://www.leadstrat.com/resources-strategic-equation-for-success.html>
8. Townsend, Robert. Up the Organization. San Francisco, CA: Jossey-Bass, 2007. Page 150

Embedded with Facebook

DoD Faces Risks from Social Media

Capt. Kenneth N. Phillips, Marine Corps Tactical Systems Support Activity
LT Aaron Pickett, Navy Information Operations Command
Simson Garfinkel, Naval Postgraduate School

Abstract. U.S. service members are increasingly jeopardized by information posted on social network websites. While some of the most damaging information comes from spouses and other non-official sources, other information comes from the use of social media by the DoD because non-public, secure channels for questions and feedback do not exist. Other problems arise from the conflict between the DoD's desire to promote its mission by distributing information to a world-wide audience and the ability of adversaries to misuse that information. We have conducted a study of information posted on Facebook and other social media websites and have determined that it is relatively easy to correlate the DoD official records with online profiles, allowing the targeting of specific warfighters. We summarize several cases in which the public disclosure of information led to mission compromise and suggest ways for improving current policy and practice.

Introduction

U.S. service members are increasingly jeopardized by information posted online by the DoD, by friends and family, by other service members, and by themselves. Information posted online can be used to target service members and their families for crime, retribution, or terrorism. Online postings can also leak sensitive information about tactics or capabilities, and can even compromise specific operations.

These risks are not hypothetical: terrorist publications have advocated collection of information from Facebook [1]; in March 2010 an Israeli raid had to be canceled because a soldier posted the details of the raid to his Facebook page [2]; and there have been persistent reports of military members being targeted by identity theft rings [3].

It is not clear how the DoD should respond. Certainly DoD Operations Security prevents direct security compromises such as publishing the time and locations of planned attacks against our adversaries.[1] But much of the most damaging information published today does not come through official channels. Attempting to regulate a spouse posting to an online support forum the location of her husband in Afghanistan would pose obvious First Amendment issues.

The DoD is better positioned to limit the disclosure of personal information on DoD websites—for example, by limiting the posting of names and photographs. But such attempts to restrict the flow of information will have an adverse impact on recruitment, public affairs, and diplomatic efforts. Currently the trend has been to embrace openness, despite the risk.

There are also strong reasons within the DoD to encourage the use of social media. Social media allows easy communication between service members and their families, improving the morale of both. These websites and services also provide excellent platforms for the informal distribution of information—even from one official source to another. Indeed, services like Facebook and Twitter are now used by the DoD in an official capacity to supplement other public affairs activities.

We argue that there is a difference between using social media for carefully controlled publications and the uncontrolled disclosure of sensitive information. To this end, we conducted an investigation of vulnerabilities that result from the intentional and inadvertent release of information about service members to the Internet between September 2009 and September 2010. We found many previously undocumented cases in which information that could be considered sensitive but which was unclassified was routinely posted by DoD personnel and their families on publicly available websites. We also developed reliable techniques for cross-correlating and fusing information between multiple freely available information sources, amplifying the risk posed by the individual disclosures.

During the course of this investigation the DoD changed its policy on Facebook and other social network websites, and now allows them to be used from official computer systems and for both personal and professional purposes. This change makes the results of our study even more important.

We believe that the new, relaxed policy needs to be accompanied by a systematic examination of information that the DoD is publishing to the Internet through both official and informal channels. DoD personnel need to understand the ability of our adversaries to integrate multiple releases of apparently innocuous information into a form that can compromise operations and personnel. Finally, service members and their dependents need to understand risks and the need for appropriate conduct.

Embedded with Social Media

Today Facebook is the world's dominant social network site. Facebook boasts over 600 million active users, half of whom check the site on any given day. According to Facebook these users share more than 30 billion pieces of content and spend over 700 billion minutes on the site each month [4].

Facebook is also the most popular social network site for DoD personnel. Using our techniques for correlating official DoD records with directories on Facebook, MySpace, and LinkedIn, we determined that (at the time of the study) between 25% and 57% of DoD personnel had Facebook accounts, between 22% to 48% of DoD personnel had MySpace accounts and 11% to 18% had LinkedIn accounts [5]. These numbers have likely increased over the past year with the continued growth and acceptance of social media sites.

In February 2010, the DoD updated its policy regarding the use of social media sites [6], directing that the Non-classified Internet Protocol Router Network (NIPRNET) be configured to allow access to social media, e-mail, instant messaging, and other Internet-based applications not controlled by the DoD or Federal Government. The new policy also allows for official uses of social media sites that are not related to public affairs and directs that all external official presences on the Internet be registered on <http://www.defense.gov>.

The DoD itself maintains official sites on Facebook, Flickr, Google Buzz, Twitter, UStream, and YouTube, along with the DoDLive Blog. All of the DoD services, including the National Guard and Coast Guard, have an official presence on Facebook, Twitter, Flickr, and YouTube. Numerous high-ranking leaders within the DoD have their own Facebook pages and are aggressively using social media for recruiting, public relations, and information dissemination. For example, the Chairman of the Joint Chiefs of Staff's page[2] has over 15,000 individuals listed as "liking" the page.

The Army, Air Force, and Marine Corps have also published guidelines[3] for service members who choose to use social media sites in an unofficial or personal capacity. The Air Force and Marine Corps guidelines help service members understand what is and what is not appropriate to post online. They also provide general recommendations for the privacy settings that members use on social media sites and remind service members that content posted online can be seen by anyone. The Army guideline provides details on specific social media sites on which the Army maintains an official presence and encourages soldiers to participate in these sites as a way of spreading positive publicity about the Army.

Deployed units are using sites such as Facebook and Twitter to share photographs and newsletters and to release official information [7]. Individual service members use Facebook and other sites to stay in contact with loved ones during deployments. Family members use these sites to keep their deployed service members informed about happenings at home and to let friends and extended family know about what is happening with their service member.

In August 2010, the Navy released an All-Navy message specifically addressing the use of Internet-based capabilities, including social network sites such as Facebook. The guidance warns service members to be careful about using third-party applications on social network sites, encourages them to learn about and use the privacy settings available on social media sites, and reminds them to be thoughtful about who they allow to access their social media profiles. The ALNAV also warns service members about the potential for criminals to use personal information posted on the Internet for identity theft [8].

Social Media Risks and Exploitation

With all of the activity taking place on social network platforms, there are bound to be leaks of sensitive information. These leaks can occur in two ways. First, a specific sensitive item might be inadvertently posted in an online forum where an adversary exploits it. But information can also be released in small bits that are later collected and correlated. Adversaries can then fuse this data to develop a more complete profile.

Potentially harmful leaks include:

- **Locations and dates of deployments**
- **Details about pending operations**
- **Identifying photos of service members**
- **Identities and location of service members' families and friends**
- **Locations of sensitive facilities**
- **Impending policy changes**
- **Non-public details of military capabilities**

These risks are not theoretical. A post on a jihadist website instructs followers to gather intelligence about U.S. military units and the family members of U.S. service members, including "what state they are from, their family situation, and where their family members (wife and children) live," and to "monitor every website used by the personnel … and attempt to discover what is in these contacts" [1].

These risks to security do not come only from adversaries attempting to collect information, but also from inadvertent posts by one's own forces. Israeli Defense Forces (IDF) postponed an operation in March 2010 after a soldier posted the location and time of a planned raid on his Facebook page [2]. In a separate instance that took place in July 2010, it was revealed that Israeli soldiers who had served at a secret IDF base had set up a public Facebook group meant for veterans of the base. Members of the group had uploaded photos of themselves inside the base. A reporter inadvertently admitted to the group copied posts and photos from the group's "wall" to his own computer [9]; quotations from the posts were later published.

While not as directly revealing as the information distributed in Israel, the DoD routinely publishes personally identifying information of service members including high-resolution photographs, name, rank, promotion dates, occupational specialty, and unit affiliations. Until a recent policy change by the Office of the Secretary of Defense [10], the last four digits of a service member's Social Security Number could be posted on public webpages. The new policy, issued shortly after our research was distributed within the DoD, called attention to the problem and its implications. These details can be combined with other publicly available records to reveal more sensitive details.

Internet queries based on disclosed information can provide home address, family status, the identity of family members, and other sensitive information. Furthermore, identifying details provided by the DoD can be used to uniquely identify and target accounts belonging to service members. This can be accomplished by matching names and photographs, or by checking for membership in Facebook groups associated with military units or specialties. It may also be possible to deduce a service member's birth year from their date of rank (since most officers are commissioned soon after college, and promote at regular intervals), and match that with biographical information on a Facebook profile. We believe that this poses a risk to service members, their dependents, and operational capabilities.

During World War II, Americans were advised not to repeat military information that they might have learned due to association with friends and families—"loose lips sink ships." These lessons are now long forgotten, as Example 1 readily confirms.

High-resolution photographs available from DoD press releases and Facebook profiles pose a special risk to U.S. forces. For one, they can be used to build biometric databases used to covertly identify these individuals years after the original photograph is released. Location-based services and geo-tagging of photographs pose yet another risk. A photograph snapped with a cell phone camera and posted to a social networking website or e-mailed to a distribution list can also inadvertently reveal the graphical location of their homes, workplace, or even sensitive locations, since many cell phones now embed geographical location within digital photographs.

Example 1: Actual posts (anonymized) from Facebook pages belonging to DoD personnel, found with a simple search

"DEPRESSED....COUNT DOWN in 32 days my better half will deploy to Afghanistan. What to do now? "

"family and friends a moment of your time to pray for my nephew chris b******, he is leaving to Afghanistan for a year of duty with the army national guard. He will deploy on august the 10th. Thank-you all."

"Please keep our family in your prayers as both of my brother deploy to Afghanistan tomorrow at 11 am....."

"To all my friends and family. Tonight say a prayer for 1-66 armor 4th infantry. Tonight will be there last night state side, as they deploy to Afghanistan."

"Dear Lord, Please keep My Husband, My Son, & their fellow Soldiers safe- and give me & our Family strength these next (very long) 12 months! "

"I want to thank those that attended the Send-Off party for my husband MAJ Doug P***** and my son, SGT Mitchell S****** as they prepare to deploy to Afghanistan in 10 days! "

Many social network users leave their profile privacy settings open to the public, allowing any web user to view their personal information. This personal information can be even more damaging if combined with profile information from family members and friends. In February 2010, Pete Warden created a script that downloaded 215 million public profile pages from Facebook, including 120 million from U.S. users. He planned to make the profile data available to academic researchers, but deleted it after Facebook threatened a lawsuit [11]. Six months later, security researcher Ron Bowes wrote a script that downloaded the names and profile URLs of 171 million Facebook users; he then made the downloaded information freely available over the Internet [12] before Facebook could intervene.

Just as damaging as the content of the individual posts is the identifying information associated with them. When this information comes from Facebook it is frequently accompanied with the true name of the person who posted it–the use of fake names or aliases is a violation of Facebook's terms of service and can result in account termination. Facebook also frequently displays that individual's friends and in some cases, where that person lives, works, and spends their time.

All of this information can be used by adversaries to improve targeting of U.S. forces and their families. The targeting of service members and their families is not unprecedented: one year after the Vincennes accidentally shot down an Iranian civilian airliner in 1988, a van belonging to the ship's former commanding officer was fire bombed in an apparent retaliatory attack.

Being able to search for results like this also makes it easy for would-be identity thieves to find out when a service member will be away, making them more vulnerable to identity theft. It's difficult for warfighters to monitor their credit when they are in a warzone.

With the relaxation of the DoD's policy on social media, commands have started using Facebook and other social media sites to share information with members and their families. As such, the DoD should be specifically concerned with the use of Facebook as an open forum for personnel and family members to ask questions related to orders and personnel records.

It is frequently not obvious to users of these pages that information posted is visible to the world and not restricted to the intended audience. Such questions potentially reveal details about service members, families, and troop deployments. Individual postings might seem harmless, but they can be useful to adversaries if they are combined with other posts, the identities of the posters, and information gathered using other methods.

In our review of Facebook, we found specific examples on command-sponsored Facebook pages that raise concern; they are shown in Example 2.

Example 2: Facebook posts that show evidence of deployments

"About 3 weeks ago we received verbals to Lemoore. We are currently stationed in Atsugi, Japan. I am in need of a early family member return because our rotation date to leave here is in mid November and I am pregnant and due November 25th."

"I am also currently awaiting orders but to ECRC NFLK fwd Afghanistan and I am currently in Guam."

"I already have PCS orders for a GSA in Aghanistan, I report to NMPS in December when should I receive my Temadd orders for my assignment and training. I saw in my orders that they should be release along side my PCS orders. I was told 60 days before I transfer from a few people. Is this right?"

Even if a Facebook group could be restricted to vetted members of the command, their dependents, or close friends, it is important to realize that Facebook's servers are not operated by the DoD. Information stored in these servers is available within Facebook to various programmers, system administrators, and others–many of whom may not be U.S. citizens, and may not even reside within the United States. Unlike DoD servers, which rely on encryption to transmit sensitive information over the NIPRNET, Facebook is generally accessed without encryption.

Facebook and other social network sites do not require identity verification prior to creating an account, which makes it easy for an adversary to impersonate an account or create a fictitious account, then befriend unknowing targets. Security consultant

Thomas Ryan set up a Facebook profile for a fictitious 25-year-old woman working at the U.S. Navy's Network Warfare Command. Within a month, the profile had over 300 contacts from within the U.S. defense and intelligence communities, an invitation to speak at a security conference, and a request to review a technical paper by a NASA researcher [13]. One military contact of the fictitious female even revealed details of take-off times for military helicopter flights in Afghanistan [14]. Ryan was also able to gain access to e-mails and one person's bank account information by making use of details published on personal profile pages to guess the answers to "secret questions" that are used as back-up authentication when a user forgets a password [13].

Already enemy organizations have used social networks to obtain intelligence. In Israel, for example, military intelligence officers were ordered to close their Facebook accounts after it was discovered that some had been "friended" by Hizbullah operatives posing as Israeli women for the purpose of gaining access to personal information [9].

Another important security problem with Facebook is the use of so-called "cookie authentication," which allows an adversary to impersonate legitimate Facebook users and gain extended unauthorized access to a Facebook account by capturing a Facebook "cookie" from an unsecured wireless network or from a public computer. Software is now widely available that gives the attacker an easy-to-use web-based interface of the cookies that have been captured; simply clicking on a user name allows the attacker to compromise any of the linked Facebook accounts at will.

Facebook allows third-party developers to write applications that users can add to their Facebook profile. These applications frequently have unrestricted access to a user's personal data. When a user permits an application access to their profile, the application can also see the profile information of that user's friends with the same level of detail that the user can see, unless it has been specifically prohibited by the friends' privacy settings. The default settings permit this behavior.

The net result of the large membership groups, the access given to "friends," and Facebook's security model is that it is unwise to store any information on Facebook that is meant to have any form of restricted dissemination.

Recommendations

Even a casual analysis of Facebook indicates that a significant amount of information is being posted that could easily be used against U.S. interests. This a growing problem that needs to be addressed.

Social media such as Facebook increasingly plays an important role in personal communication, entertainment, political discussions, and even the dissemination of official information. The DoD has already decided that it makes more sense to embrace social media than to attempt a futile ban. Indeed, if the DoD were to abstain from the new media in an official capacity and ban its use, it is likely many of the conversations would remain active in unofficial capacities. But as our work shows, social media is creating real risks and vulnerabilities for the DoD. Given the scale of the problem, the most effective near-term solution we see is education.

Service members must be taught to understand the risks involved in posting personal information on the Internet, not only to themselves, but to their units and families. They need to be informed about the different levels of privacy available on social network sites and the implications of each level. They also need to understand that the privacy level they select is not a guarantee of privacy. There have been leaks of private information in the past and there are bound to be more leaks of private information in the future. The reality is that any information that is posted to a social media website may readily become available to the public at large–access controls are not effective.

The DoD needs to consider ways to make service members and their families as safe as possible when using social media. One way to do this is to provide specific guidelines of how individuals can use these services safely, as well as examples of how lax practices may make us vulnerable to Open Source Intelligence collection by our adversaries.

Recently there have been some efforts to educate the services. For example, the Department of the Navy Chief of Information produced a briefing with "Recommended Facebook Privacy Settings."[4] The briefing explains how Facebook makes money by showing targeted advertisements. The materials rightfully warn that anything stored in Facebook could be made public–manipulating the privacy settings is no guarantee of preserving privacy. Nevertheless, the briefing does give specific recommendations on how to set Facebook's complex privacy settings. Keeping materials such as this up-to-date will be a challenge given Facebook's tendency to make rapid and significant changes to both its user interface and its underlying privacy policy.

Other services are taking similar measures. The Marine Corps is incorporating education on social media use into annual operational security and information assurance training [15]. The Army Memorandum on the responsible use of Internet-based capabilities [16] warns that the use of social networking sites by Army personnel provides adversaries with the opportunity to gather personal information that can be used to directly target Army and DoD personnel.

Educating the service members is not enough. We have seen posts by spouses, children, parents, and friends that revealed details about the location or deployment dates of their service member. By itself, this information might seem harmless, but when it is put together with information from other posts and other sources, it can become dangerous. An adversary could easily determine the address of a service member's family based on their name and the location information in their profile. Then they can find out the location of the children's schools or daycares. An innocent post by a wife that her husband is halfway through his deployment in Afghanistan can alert an adversary that the family might be extra vulnerable to an attack. To this end, the Army directs that personnel discuss the proper use of social media with family members using a guide[5] specifically tailored to family members.

Technology can also be of help. For example, the website ReclaimPrivacy.org operates a "privacy scanner" that allows individuals to scan their own Facebook privacy settings. Google has a plug-in for its Gmail service that detects attempts to send e-mail that one might later regret. Similar technology could be developed by the DoD to protect privacy, strip location information from photographs, or scan messages and postings for sensitive information.

Nevertheless, one of the fundamental problems with today's social networks is the lack of authenticated identity. When a service member receives a "friend" request from an old friend or

classmate, it can be difficult, if not impossible, to authenticate that request. But such authentication is important with today's social networks that provide more information to "friends" and "followers" than to outsiders.

One way around this problem would be for the DoD to provide an alternate social network site for DoD members and their families. Such a site could allow family members to communicate with service members and with each other in a more secure setting that is not available to the general public. Membership to the site could be controlled and restricted to only service members and those they invite to the site. More stringent privacy settings could be provided and enforced so that profiles and posts are not visible outside of directly connected relationships.

ABOUT THE AUTHORS

Capt Kenneth Phillips is a recent graduate of the Naval Postgraduate School in Monterey, California where his master's thesis explored correlating public DoD records with social network websites. He currently serves as the SATCOM Project Support Officer at the Marine Corps Tactical Systems Support Activity.

Capt Kenneth Phillips
MCTSSA, Box 555171
Camp Pendleton, CA 92055-5171
E-mail: kenneth.n.phillips@usmc.mil

LT Aaron Pickett is a U.S. Navy Information Warfare Officer currently assigned to Navy Information Operations Command (NIOC) Suitland. He graduated from LeTourneau University in 2002 with a BS in Computer Science and Engineering, and from the Naval Postgraduate School in 2010 with a MS in Computer Science. His past assignments include NIOC Hawaii and instructor duty at Naval Nuclear Power Training Command.

LT Aaron Pickett
NIOC Suitland
4251 Suitland Road
Washington, DC 20395-5720
E-mail: aaron.pickett@navy.mil

Simson L. Garfinkel is an Associate Professor at the Naval Postgraduate School in Monterey, California. His research interests include computer forensics, the emerging field of usability and security, personal information management, privacy, information policy and terrorism.

Simson L. Garfinkel
Naval Postgraduate School
Monterey, CA
E-mail: slgarfin@nps.edu

REFERENCES

1. Phil Ewing. The terror threat at sea. Navy Times Scoop Deck Blog, December 31 2009. <http://militarytimes.com/blogs/scoopdeck/2009/12/31/the-terror-threat-at-sea/> Accessed April 23, 2011.
2. Yaakov Katz. Facebook details cancel IDF raid. The Jerusalem Post, March 2010. <http://www.jpost.com/Home/Article.aspx?id=170156>
3. Geoff Ziezulewicz. ID theft surges among US troops in UK. Stars and Stripes, November 18 2008. <http://www.military.com/features/0,15240,179476,00.html>
4. Facebook statistics. Facebook Press Room, 2010. <http://www.facebook.com/press/info.php?statistics> Accessed April 23, 2011.
5. Kenneth N. Phillips. Correlating personal information between DoD411, LinkedIn, Facebook, and Myspace with uncommon names. Master's thesis, Naval Postgraduate School, 2010.
6. DoD Directive-Type Memorandum 09-026 responsible and effective use of Internet-based capabilities, February 2010. <http://www.defense.gov/NEWS/DTM%2009-026.pdf>.
7. James Dao. Military announces new social media policy. New York Times At War Blog, February 2010. <http://atwar.blogs.nytimes.com/2010/02/26/military-announces-new-social-media-policy>. Accessed April 23, 2011.
8. ALNAV (All Navy) 057/10 Internet-based capabilities guidance, August 2010. <http://www.public.navy.mil/bupers-npc/reference/messages/Documents/ALNAVS/ALN2010/ALN10057.txt>
9. The Media Line. There are things we'll never know. The Jerusalem Post, July 2010. <http://www.jpost.com/Israel/Article.aspx?id=180838> Accessed April 23, 2011.
10. Office of the Secretary of Defense. Policy Memo 13798-10, "Social Security Numbers (SSN) Exposed on Public Facing and Open Government Websites." November 23 2010.
11. Jim Giles. Data sifted from Facebook wiped after legal threats, March 2010. <http://www.newscientist.com/article/dn18721-data-sifted-from-facebook-wiped-after-legal-threats.html> Accessed April 23, 2011.
12. Ron Bowes. Return of the Facebook snatchers. Internet Blog, July 2010. <http://www.skullsecurity.org/blog/2010/return-of-the-facebook-snatchers>. Accessed April 23, 2011.
13. Shaun Waterman. Fictitious femme fatale fooled cybersecurity. The Washington Times, July 2010. <http://www.washingtontimes.com/news/2010/jul/18/fictitious-femme-fatale-fooled-cybersecurity> Accessed April 23, 2011.
14. Michael Cosgrove. US military and security fooled by Internet 'friends' hoax. Digital Journal, July 2010. <http://www.digitaljournal.com/article/295079>. Accessed April 23, 2011.
15. MARADMIN (Marine Administrative Message) 181/10 responsible and effective use of Internet-based capabilities, March 2010.
16. SAIS-GKM (Assistant Secretary of the Army, Information Systems – Governance, Acquisition & Chief Knowledge Office Memorandum) Memorandum, responsible use of Internet-based capabilities, 2010. <http://www.slideshare.net/DepartmentofDefense/army-official-social-media-policy>

NOTES

1. The DoD Operations Security (OPSEC) Program Manual 5205.02-M (November 3, 2008) describes a step-by-step approach for identifing and mitigating OPSEC risks, much of which relies on education, training and general awareness.
2. http://www.facebook.com/admiralmikemullen
3. http://socialmedia.defense.gov
4. http://www.doncio.navy.mil/ContentView.aspx?ID=1818
5. https://www.us.army.mil/suite/page/589183

Browser User Interface Design Flaws

Exploiting User Ignorance

Aditya K. Sood, Michigan State University
Richard J. Enbody, Ph.D., Michigan State University

Abstract. A browser is considered to be a functional window to the Internet. It is interface software that serves as a communication medium between the users and the Internet. Sophisticated attack patterns and design flaws in browsers pose serious threats to user security, privacy, and integrity. Recent advancements have shown that browser User Interface (UI) design flaws catalyze the vulnerability exploitation. This paper sheds light on the design flaws in Graphical User Interface (GUI) components of browsers that are exploited by the attackers to trick users to perform rogue operations. In most of the cases, the user is unaware of the attack that results in stealth operations. Thus, user ignorance plays a critical role in successful exploitation of the design flaws.

Introduction

GUI plays an instrumental role in the success of any browser. GUI enables user control and improves interaction with the browser. GUI is considered a part of the user trust model for all types of software including browsers. Apart from the main browser window, GUI in browsers includes notification bars, status bars, address bars, download dialog boxes, HTTP authentication dialog, and browser objects such as frames, buttons, etc.

Users interact with the GUI components in their routine life jobs. GUI flaws are considered design bugs in which an attacker can circumvent the normal functioning of the browser by running malicious JavaScript. Primarily, GUI bugs in browsers are mostly exploited by spoofing [1] and clickjacking attacks [2]. Spoofing attacks are those kinds of attacks that tamper the UI component of software in order to fool users into performing false operations by exploiting their ignorance. They fail to differentiate between the real and manipulated objects in software. Clickjacking attacks fall into the category of UI redressing attacks in which an attacker embeds a hidden UI object such as buttons, frames, etc. to execute stealth functions that are binded to a real object. For example, an attacker can easily place a hidden button over the real button in a browser window that executes a malicious function when a user clicks it.

Basically, spoofing and clickjacking attacks aim at tampering with and manipulating the functional operations of various browser GUI controls. Apart from this, such attacks exploit the user ignorance to a great extent because users are not able to differentiate between the real GUI object and vice versa. Successful GUI attacks depend a lot on the user awareness about the browser controls and their integrity. It is a major concern because exploitation of GUI design flaws can severely impact the user trust thereby resulting in the loss of integrity. This paper discusses design flaws in browsers related to GUI components and how they are exploited by tricking the user.

HTTP Authentication Dialog Spoofing

Many browsers require HTTP-based authentication in which users have to provide a set of credentials to access the resources. In general, if a resource is protected, the server sends a particular HTTP response to the browser based on which the browser initiates a dialog authentication process. It is one of the main characteristics of browsers to handle HTTP authentication. Every single HTTP authentication process has a realm value associated with it. In general, the realm value is a string that shows the domain name on which resource is protected. The realm value also provides a user supplied string for identity purposes. A user can check the domain name and provide his credentials to gain access to the server. However, recent vulnerabilities have shown the fact that it is possible to manipulate the authentication dialog box. Users are unable to differentiate among the origins of authentication dialog. A dialog box may look real and authentic but it can be spoofed. This type of flaw in browsers results in the stealing of user credentials without users being aware of the reality. For example: Internet Explorer and Google Chrome inherit this design flaw. A serious design flaw in Google Chrome [3],[4] is that an authentication dialog can be completely spoofed and users are not able to distinguish the difference. A spoofed authentication dialog box is presented as in Figure 1.

Figure 1: Spoofed authentication dialog box in Google Chrome

The spoofed authentication dialog box bedazzles the user. However, it has been noticed that a number of users fall into this trap and provide their authentication credentials as per the realm value shown in the dialog box. This design flaw persists because browsers are not able to handle the realm value passed as a parameter to the authenticated HTTP response header and render it directly in the dialog box. Most browsers do not handle the realm value in an appropriate manner, allowing spoofing attacks.

URL Obfuscation Flaws

URL obfuscation is one of the most notorious problems noticed in browsers. Continuous efforts have resulted in correction of this design problem in a number of browsers such as Mozilla, Internet Explorer, etc. However, browsers such as Google Chrome still inherit this design bug. In 2008, a design flaw [5] was released in Google Chrome that still persists in recent versions [6]. URL obfuscation is a trick that plays around the designing of URLs with certain meta characters in order to confuse browsers as well as users so that they can be redirected to malicious domain. This is a browser design flaw because browsers are not able to render the URLs appropriately thereby resulting in unauthorized redirection. As a result, the browser can be redirected to a malicious domain that is ready to serve malware.

There can be many combinations based on this pattern. It depends on the inherent design of the browser in interpreting a URL. In general, good practice requires that browsers should raise a warning about the obfuscation in a URL and should be smart enough to present a user with an appropriate choice. Primarily, the user thinks that a destination website is Google.com, but in reality, the user is redirected towards yahoo.com. An obfuscated URL is shown in Figure 2.

In Figure 2, Google Chrome is redirected towards yahoo.com instead of raising a warning or going to google.com. A similar test on Mozilla raises a warning about the URL obfuscation as presented in Figure 3:

After a lot of discussion, Mozilla introduced a security check to show concern with URL obfuscation flaws in browsers.

Manipulating Browser Status Bars

Browser status bars are used to present the active state of links when a user clicks a hyperlink on a webpage. In general, status bars represent the status of hyperlinks. The mindset behind the design of the status bar is that a user can see the authenticity of domain names and hyperlink. Basically, a user believes that the status bar displays the domain name in the form of a URL and the browser redirects to that page upon clicking. Attackers have exploited this design flaw by spoofing the status bar with JavaScript calls such as window.location or window.href to fool users. However, the URL obfuscation trick can also be used to spoof the status bar. An issue was raised in Internet Explorer [7] about the problem in the status bar. When considering spoofed HTML code, Internet Explorer 7 does not appropriately render the information in the status bar whereas Internet Explorer 8 does not even show any information in the status bar when a mouse is pointed over a hyperlink. This a serious issue because it is the only way a normal user can scrutinize the authenticity of a hyperlink. Figure 4 shows code that is used to spoof the status bar in Internet Explorer.

Figure 2: URL Obfuscation in Google Chrome

Figure 3: URL Obfuscation Warning in Mozilla Firefox

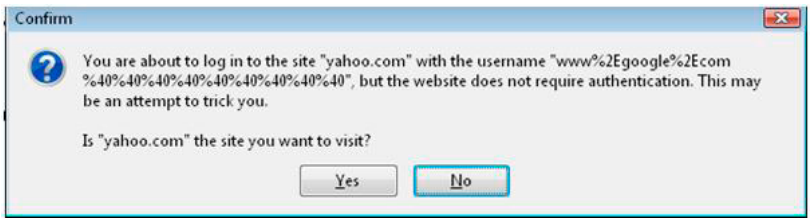


Figure 4: Custom HTML Code to Spoof Internet Explorer's Status Bar

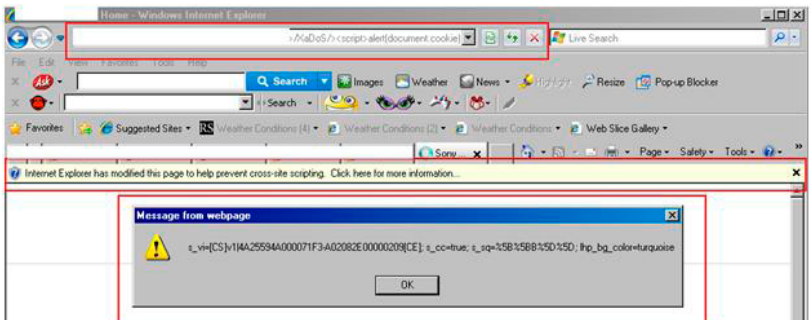
An active Internet Explorer test has been structured here [8]. This is a simple but generic problem in most of the browsers and it is used quite often by attackers to trick users and create a false sense of security.

Cross Site Scripting Attack Notification Bars – Bypassing Filters

With the advent of new browser security protection mechanisms, reflective Cross Site Scripting (XSS) filters have become a part of the browser architecture. It is an inbuilt protection mechanism that raises XSS attack notification bar for reflective XSS attacks and neutralizes them completely. This is the actual motive behind the designing of XSS filters. However, completely relying on filters as a fool proof protection against XSS attacks creates a false sense of security. The XSS filters in browsers are not well developed and can be bypassed easily to execute successful XSS attacks. Primarily, a user believes that now the browser is secure because of the presence of XSS filters but attackers can exploit the design problem in XSS filters to exploit the trust of users. For example, Internet Explorer released a built-in XSS filter with Internet Explorer 8, but it can be bypassed easily and no notification alert is raised. Moreover, certain stealth XSS attacks were successfully executed in In-

ternet Explorer. However, Internet Explorer's XSS filter raised a notification warning but was not able to sanitize the XSS attacks appropriately. This type of behavior shows the inherent weakness in client-side XSS filters. Moreover, NoScript is considered a very good extension of Mozilla that prevents reflective XSS attacks. However, there are certain bypasses that have been released in it. The good point about this filter is that one can find a lot of updates of this extension. Figure 5 shows a potential attack against the XSS filter in Internet Explorer.

Figure 5: Successful bypass even after XSS notification



This attack simply projects how the design issues in XSS filters result in exploitation of vulnerability.

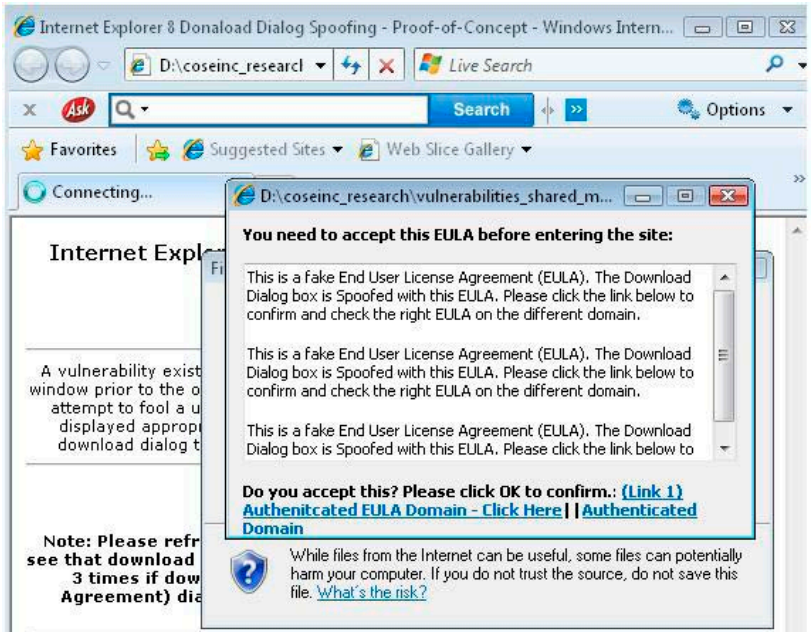
Download Dialog Box Spoofing

Browsers use a download dialog box in order to download a file from a server. This process acts as a notification to the user about the characteristics of the file. The download dialog box is displayed when a user clicks a hyperlink to download a specific file. It is a type of GUI displayed to the user for raising an alert. Attackers are spoofing download dialog boxes to trick users into downloading malicious files instead of authorized files. This attack is triggered on a wide scale to infect user machines with malware. Recently performed tests on Internet Explorer have shown that it is possible to overlap the download dialog box with an unauthorized pop-up window which restricts the functionality of the download dialog box.

Primarily, the overlapped pop-up window forces the user to click some malicious links embedded in it. The pop-up window actually locks the authorized download dialog box and the user fails to download the file directly. This attack is implemented in order to force a user to interact with the rogue pop-up window. In other words, it is a design bug in Internet Explorer that fails to differentiate between the download dialog box and a rogue pop-up window. Figure 6 shows the spoofed download dialog box in Internet Explorer 8.

In the Figure 6 screenshot, a fake End User License Agreement (EULA) pop up window overlaps the authorized download dialog box. This fake EULA window is embedded with malicious links and it locks the download dialog box completely. This attack forces the user to interact with a EULA window prior to downloading the file. In general, users are not aware of these design problems and spoofing tricks which help an attacker to launch attacks successfully. The figure clearly shows one of the serious design bugs in graphical user components in browsers.

Figure 6: Spoofed Download Dialog Box



Clickjacking Browser Interface

Clickjacking [9],[10] is a UI redressing attack where an attacker executes malicious functions by playing around with browser UI components. The aim of this attack is to steal sensitive data and extract information about a user's activities in a stealthy manner. Primarily, this attack uses two major UI components in a browser–frames and buttons. The term clickjacking itself points to hijacking mouse clicks in a browser window. In general terms, an attacker designs a transparent UI component such as a button and makes it hidden. When a legitimate user performs a mouse click in a browser window, the hidden button is clicked and it executes the backend command designed by the attacker to perform rogue functions. This attack is considered one of the most sophisticated attacks.

Solutions

In order to prevent these attacks, here are some measures that can result in mitigating the adverse attacks to some extent, but it is hard to guarantee foolproof solutions:

1. An appropriate browser-based filter should be used while surfing the Internet. For example: NoScript [11] is a good choice. It only works on Mozilla Firefox but it has some built-in capabilities to take control of certain UI redressing attacks such as clickjacking.
2. Browsers should be upgraded regularly and security recommendations must be applied in a timely manner. Most browser software vendors such as Microsoft, Apple, and Mozilla release security advisories about potential vulnerabilities. These security advisories contain an updated fix and patch that should be installed in order to upgrade the requisite browsers. However, if automated updates are enabled, the system is updated regularly against potential threats. A user can also download individual security updates manually from vendor websites.
3. Browser design requires a significant amount of change in the way UI components are handled. However, it becomes hard for the vendors to change UI on a regular basis. This is a para-

dox in the field of browsers, but vendors should take appropriate steps to secure the design interface.

4. Users should not visit those pages that they are not sure of. Sometimes, being paranoid is a good way to be secure. Always think twice about what you click. There are certain client-side browser filters available that help users substantially to make smart decisions if a potential threat is detected. For example, the NoScript plug-in works as an inline component with Mozilla Firefox to strengthen security. It enables the user to surf in a secure manner and raises notification against insecure objects and attacks such as XSS. Other browsers such as Internet Explorer come with built-in client-side protection against XSS attacks. Thus, potential combinations of client-side filters and user awareness can lower the exploitation ratio of vulnerabilities.

5. Users should be aware of the basic attacks on the Internet that can help them in understanding exploitation attempts. There are a number of websites such as Threatpost [12] , SecurityFocus [13], and Register, [14] etc. that provide substantial information about new research and attacks.

6. Websites should use frame-bursting scripts [15] to avoid framing of websites. This process is followed in order to avoid loading a website into a frame which is used by a third party. Frame-bursting scripts remove the frame when an attacker tries to load the target website into a frame. This technique avoids the hidden frames used in conjunction to launch clickjacking attacks.

7. A good use of declarative security in HTTP response headers [16,17,18] can circumvent some attacks. This is a potential step in defeating clickjacking attacks. Restricting frames [19] and running them in sandbox is also a good practice.

Conclusion

We discussed a number of cases of UI design flaws and how they are exploited. During the course of this paper, we have realized that UI is a very critical component of browsers. UI is important because it provides direct functionality to users and helps them to make decisions quickly. However, if UI design flaws are exploited, it becomes much easier to launch attacks as discussed previously. Of course, user ignorance and inappropriate knowledge enhances the chance of exploitation. These design flaws are inherited in browsers to a great extent and it is hard to remove them completely. It is hard to ensure a foolproof solution, but if a reliable set of protective measures is applied, impact can be moderated to some extent.

ABOUT THE AUTHORS

Aditya K. Sood is a security researcher, consultant, and Ph.D. candidate at Michigan State University. He has worked in the security domain for Armorize, COSEINC, and KPMG and founded SecNiche Security. He has been an active speaker at conferences like RSA, Toorcon TRISC, Hacker Halted, EuSecWest, ExCaliburCon, EuSecwest, XCON, OWASP AppSec, Security-Byte, CERT-IN and has written content for HITB Ezine, ISACA, ISSA, Hakin9, and Usenix Login.
E-mail: adi_ks@secniche.org
Phone: 517-755-9911

Dr. Richard Enbody is an Associate Professor in the Department of Computer Science and Engineering, Michigan State University. He joined the faculty in 1987 after earning his Ph.D. in Computer Science from the University of Minnesota. His research interests are in computer security, computer architecture, web-based distance education, and parallel processing. He has two patents pending on hardware buffer-overflow protection, which will prevent most computer worms and viruses. He recently co-authored a CS1 Python book, The Practice of Computing using Python.

REFERENCES

1. Wu, Yongdong, Ma. Di and Sheng, Chnag Xu. "Browser Spoofing". <http://dspace.lib.fcu.edu.tw/bitstream/2377/1421/1/ce07ics002002000204.PDF>. 2002
2. Law, Eric. Combating Clickjacking with X-Frame options". <http://blogs.msdn.com/b/ieinternals/archive/2010/03/30/combating-clickjacking-with-x-frame-options.aspx>. 23 March 2010
3. Sood, Aditya K. "User Interface Security- Google Chrome: HTTP AUTH Dialog Spoofing through Realm Manipulation ". <http://zeroknock.blogspot.com/2010/08/google-chrome-http-auth-dialog-through.html>. 23 August 2010
4. Sood, Aditya K. "Google Chrome: HTTP AUTH Dialog Spoofing through Realm Manipulation (Restated) ". <http://www.securityfocus.com/archive/1/513243/100/800/threaded>. 23 August 2010
5. Sood, Aditya K. "Google Chrome MetaCharacter URI Obfuscation Vulnerability". <http://www.securiteam.com/windowsntfocus/6L0001FN5S.html>. 25 November 2008
6. Sood, Aditya K. "Google Chrome URI Obfuscation Vulnerability". <http://www.secniche.org/gcuri/>. 2009
7. Sood, Aditya K. "Internet Explorer 8 – Anti Spoofing is a Myth ". <http://www.secniche.org/ie_spoof_myth/>. 2009
8. Sood, Aditya K. "Internet Explorer 8 – Anti Spoofing is a Myth - Demonstration". <http://www.secniche.org/ie_spoof_myth/script.html>. 2009
9. Hansen, Robert. and Grossman, Jeremiah. "ClickJacking". <http://www.sectheory.com/clickjacking.htm>. 12 September 2008
10. Guya. "Malicious Camera Spying using Click.Jacking". <http://blog.guya.net/2008/10/07/malicious-camera-spying-using-clickjacking/>. 20 October 2008
11. Maone, Giorge. "FAQ's - NoScript Client Side Protection Filter". <http://noscript.net/faq>
12. Threatpost, "Latest Computer Security News Portal". <http://www.threatpost.com>
13. SecurityFocus, "Security News Portal Website". <http://www.securityfocus.com>
14. The Register, "Security News Portal Website". <http://www.register.com>
15. Wikipedia. "Frame Killer". <http://en.wikipedia.org/wiki/Framekiller>
16. Lawrence, Eric. "IE8 Security Part VII: ClickJacking Defenses". <http://blogs.msdn.com/b/ie/archive/2009/01/27/ie8-security-part-vii-clickjacking-defenses.aspx>. 27 January, 2009
17. Sood, Aditya K. and Enbody, Richard J. "Conundrum of Declarative Security in HTTP Response Headers – Lessons Learned". <http://www.usenix.org/event/collsec10/tech/full_papers/Sood.pdf>. 10 August 2010
18. Coates, Michael. "X-Frame Options". <http://blog.mozilla.com/security/2010/09/08/x-frame-options/>. 9 August 2010
19. Lawrence, Eric. "Using Frames More Securely". <http://blogs.msdn.com/b/ie/archive/2008/01/18/using-frames-more-securely.aspx>. 18 January, 2008

Upcoming Events

Visit <http://www.crosstalkonline.org/events> for an up-to-date list of events.

Management of Change Conference
15-17 May 2011
The Homestead Hot Springs, VA
<http://www.actgov.org/events/managementofchange/Pages/default.aspx>

TechAmerica Systems, Standards, & Technology Council (SSTC) Meeting
15-19 May 2011
Charlotte, NC
<http://www.acq.osd.mil/se/events>

Systems and Software Technology Conference
16-19 May 2011
Salt Lake City, Utah
<http://www.sstc-online.org>

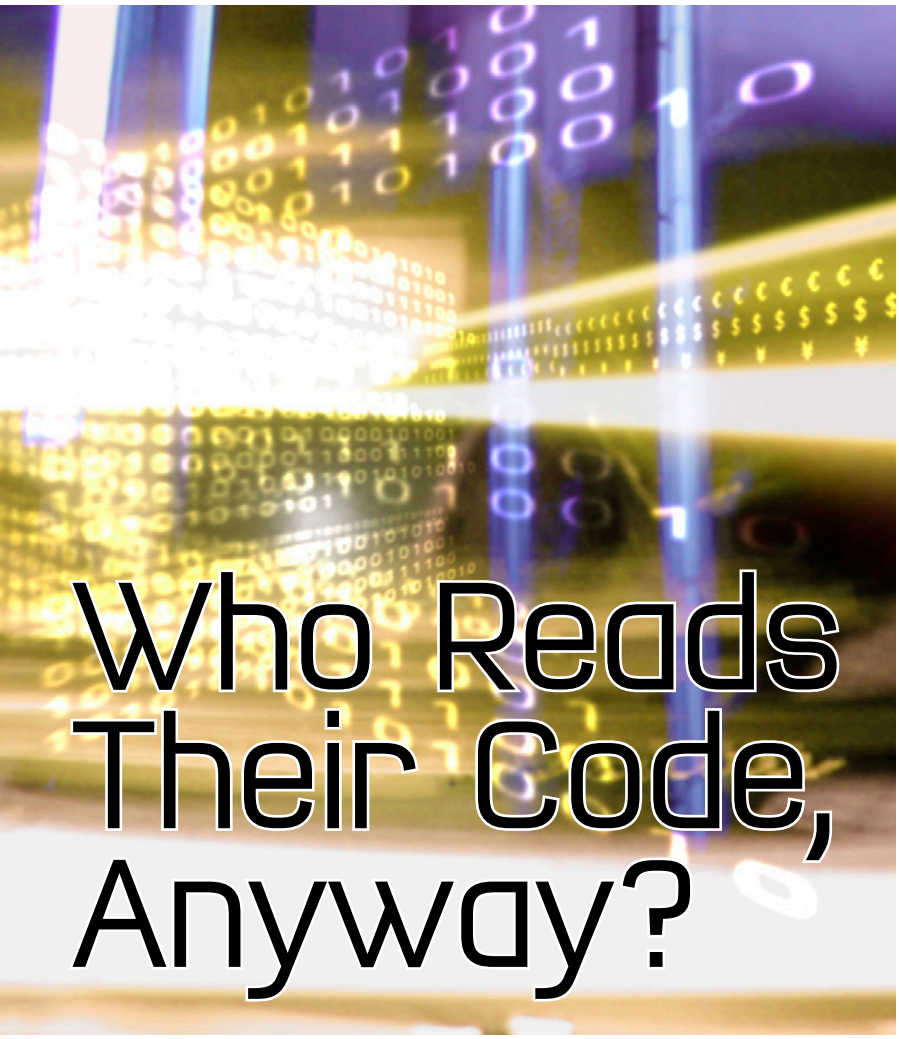
Software Engineering Institute
7th Annual Architecture User Network (SATURN) Conference
16-20 May 2011
Burlingame, CA
<http://www.sei.cmu.edu/saturn/2011>

21st Annual INCOSE Symposium
20-23 June 2011
Denver, CO
<http://www.acq.osd.mil/se/events>

Diminishing Manufacturing Sources & Material Shortages (DMSMS) & Standardization 2011
29 August – 1 September 2011
Ft. Lauderdale, FL
<http://www.acq.osd.mil/se/events>

45th Engineering & Technical Management (ETM) Conference
11-15 September 2011
St. Louis, MO
<http://www.acq.osd.mil/se/events>

14th Annual NDIA Systems Engineering Conference
24-27 October 2011
San Diego, CA
<http://www.acq.osd.mil/se/events>



Who Reads Their Code, Anyway?

My very first programming class was back in 1969. The junior high school I went to was blessed to have a Wang Programmable Calculator. It had 256 bytes of memory, a single card reader (as in "single-card reader", not "single card-reader"–80 commands could fit onto a single IBM standard 12-row 40 column card) and a paper tape reader. Programming was pretty basic, as it was meant to be used as a calculator, not a computer. Because of the limited memory and commands, you had to really work to get your programs wedged into the scarce memory. You thought in terms of efficient (albeit hard to read) code.

I was lucky, as my high school had a time-sharing terminal that allowed us to dial in to a GE computer running BASIC and FORTRAN. Yes, Virginia, GE made computers. Back in the 60s, there were eight major computer companies. IBM, the largest, was called "Snow White," followed by the "Seven Dwarfs" (Burroughs, NCR, Control Data Corporation, Honeywell, RCA, UNIVAC and GE). GE eventually sold their computer business to Honeywell. While the programs were bigger and memory not as scarce (the machine we connected to had 96 KILOBYTES of memory!!!), the slow 300-baud modems made efficient coding and debugging critical. My high school had to pay for the long-distance phone calls, so great emphasis was placed on locally desk checking before the costly phone call to upload and download the execution results. It wasn't enough to have efficient code–it had to be easy to read, and easy to debug.

Over the years, there were always tight constraints affecting how I wrote code. At one time, I was working on Contingency Operation/Mobility Planning and Execution System, Logistics Module B. The area I worked on involved calculating the efficient loading of pallets to fit onto an aircraft. This type of problem is referred to as a "bin-packing problem" and is frequently solved by using recursion–a common technique used in programming, allowing you to write programs, procedures, methods or functions that call themselves. While still a widely used and useful language, COBOL had some limitations. At the time, COBOL did not have methods nor did it allow recursion, even at the program level (both of these limitations are now allowed in modern COBOL). The problem was that the code was (and, for all I know, still is) coded using COBOL. Our solution was to simulate recursion using COBOL data structures. For a bunch of young programmers, we quickly realized that you couldn't just hack code. We spent a lot of time designing the code–creating architectural, interface, data, and modular design documents.

I could go on and on–but I know that each and every one of you who has written (or managed) coding projects have learned your own lessons. Although you might not have known it at the time, you almost certainly learned the hard way the "four pillars of software engineering" (reliability, understandability, modifiability and efficiency) as discussed in the book, *Software Engineering with Ada* by Grady Booch.

Reliable programs are important. So are efficient ones. And, given changing requirements, modifiable programs are critical. But, without understandable programs, the other three pillars are pretty much impossible. You might possibly write a reliable and efficient program that works initially, but if you can't understand the code, then attempts to modify it will certainly cause it to fail.

The best ophthalmologist I ever had was an Air Force physician who had a business card that read, "If you can't see them, you can't shoot them." Clarity is important in vision, *and* in software. I once took the Personal Software Process℠ class, and then taught it for many years. I learned the hard way that the best thing I could do to write code quickly was to write it clearly–because debugging was a very time-consuming activity. I learned the hard way that clear, easy-to-understand code is much quicker to debug and modify.

People read the code we write. The code will be around for years and years, and will be read and modified and re-read, and modified again. When it gets down to the basics, coding is a very people-based activity. To quote from the BackTalk column in the 2010 November/December issue of *CrossTalk*, "Always code as if the guy who ends up maintaining your code will be a violent psychopath who knows where you live" (attributed to Martin Golding).

After all, if you can't read it, you can't fix it.

David A. Cook
Stephen F. Austin State University
cookda@sfasu.edu

Disclaimer:

®CMMI is registered in the U.S. Patent and Trademark Office by Carnegie Mellon University.

Top 5 AWARD NOMINATIONS

2010 DoD Systems Engineering Top 5 Program Awards

NOMINATE YOUR PROGRAM

Presented to both government and industry, the awards recognize significant systems engineering achievement by teams of industry and government personnel.

- Winners must demonstrate successful implementation of systems engineering best practices resulting in DoD program success based on 2010 performance.
- Programs are considered for this award at any point in the programmatic life cycle.
- Successful applicants should have passed a sufficient number of internal milestones to demonstrate the impact of systems engineering practices.

Sponsored by the
Deputy Assistant Secretary of Defense for Systems Engineering and the National Defense Industrial Association Systems Engineering Division

Nomination packages due July 1, 2011.

Programs will be notified by September 1 of their selection.
Awards will be presented at the annual NDIA Systems Engineering Conference
San Diego, CA, October 24-27, 2011.

Visit the NDIA website to download submission instructions:
http://www.ndia.org/Divisions/Divisions/SystemsEngineering/Pages/Awards.aspx

NAVAIR

CROSSTALK thanks the above organizations for providing their support.